

4787 : PLATAFORMA E-COMMERCE MAGENTO,
MILLORA DE LA IMPORTACIÓ MASSIVA DE PRODUCTES

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Oriol Augé Gibert
i dirigit per
Ramon Martí
Bellaterra, 1 de febrer de 2012

El firmant, Ramón Martí , professor del Departament d'Enginyeria de la Informació i de les Comunicacions de la Universitat Autònoma de Barcelona

CERTIFICA:

Que la present memòria ha sigut realitzada sota la seva direcció per Oriol Augé Gibert

Bellaterra, 1 de Febrer de 2012

Firmat: Ramon Martí

El firmant, Greg Robbins , Cap de projecte i director tècnic de l'empresa Etailers

CERTIFICA:

Que la present memòria ha sigut realitzada sota la seva direcció per Oriol Augé Gibert

Bellaterra, 1 de Febrer de 2012

Firmat: Greg Robbins

Agraïments

Primer de tot, vull a la meua família tot el suport que m'ha donat en aquest projecte, sobretot en els últims mesos d'entrega. En especial vull agrair al meu pare i a la Núria el seu suport i la seva ajuda en la memòria, en la correcció i intentar entendre un món que per a ells, és desconegut.

En segon lloc a tota l'empresa, tant equip de màrqueting, equip de disseny i a l'equip tècnic, per tot el suport i ànims que m'han anat donant durant el seu desenvolupament i, en especial, a Greg Robbins per creure en el projecte, guiar-me en el desenvolupament i, sobretot, per ensenyar-me i ajudar-me a millorar en el món de l'informàtica i la programació web.

En tercer lloc, agrair a Ramon Martí la seva participació en el projecte. Per guiar-me i dirigir-me durant tots aquests mesos.

Per acabar, a tots els meus amics i coneguts que m'han animat a continuar endavant i m'han aguantat en els moments de més estrés. Per tots els cops de mà que m'han donat cada un d'ells, gràcies.

Índex

Índex.....	v
Índex de figures.....	viii
Capítol 1: Introducció.....	1
Motivació.....	1
Objectius del projecte	2
Justificació dels objectius	2
Per a què un parser de productes?	3
Per què Python i PHP?	3
Prioritat dels objectius	4
Equip del projecte	5
Què es vol aconseguir?	6
Estat de l'art	6
Plataformes de comerç electrònic (e-commerce).....	6
Per què millorar el sistema d'importació de Magento?.....	8
Estudi de viabilitat del projecte	9
Viabilitat legal	9
Viabilitat tècnica i operativa.....	10
Viabilitat econòmica	10
Conclusió	10
Planificació temporal del treball	11
Planificació	12
Descripció dels principals grups de tasques	13
Canvis en la planificació.....	16
Plantejament d'un nou model de negoci	16
Previsió de gran temps de manteniment.....	18
Poca implicació dels clients	18
Introducció de noves eines	18
Segona planificació	18
Capítol 2: Anàlisi i Disseny	21
Anàlisi.....	21
Perfils d'usuari	21

Requisits no funcionals.....	21
Requisits funcionals.....	22
Disseny	23
Introducció	23
Selecció de l'estructura de classes	24
Patró "Front Controller"	25
Diagrama de l'estructura del programa	26
Selecció del llenguatge de programació.....	26
Llenguatge de programació PHP	27
Llenguatge de programació Python	27
Necessitat d'una base de dades	28
Consultes a la base de dades de Magento	28
Perquè SOAP i no XML-RPC	29
Entorn de desenvolupament	29
Git: Control de versions pel projecte.....	30
Diagrames de seqüència.....	31
Capítol 3: Implementació.....	32
Google Docs.....	32
Passar de Python a PHP.....	32
Creació d'un mòdul Magento.....	34
Zend Framework.....	34
MVC (Model – Vista - Controlador)	35
Arquitectura de Magento.....	36
Estructura de fitxers	37
Estructura final del projecte.....	41
Idiomes, què implica?	45
Llista de comandes.....	46
Capítol 4: Proves i Aplicació del projecte	48
Vila viniteca	48
YouBuyOutdoor.....	49
Motorstoreshop	50
Seleqto	51
Altres botigues	52

Capítol 5: Conclusions	53
Objectius assolits.....	53
Línies d'ampliació	54
Valoració personal.....	55
Utilització per part dels clients.....	55
Millores respecte al sistema inicial	56
Planificació final	57
Annex: Instal·lació del projecte	60
Manual del programa: Instal·lació	60
Csv en els productes configurables	63
Exemples de missatges d'error a Google Docs	65
Bibliografia.....	66

Índex de figures

Figura 1: Tasques planificació 1	13
Figura 2: Diagrama de grantt. Cada tasca està identificada amb la seva Id.....	15
Figura 3: Escenari 1	16
Figura 4: Escenari 2	17
Figura 5: Tasques planificació 2	19
Figura 6: Diagrama de grantt plaificació 2.	20
Figura 6: Representació del MVC	24
Figura 7: Diagrama de flux de l'aplicació.....	26
Figura 8: Esquema comunicació Apache + PHP	27
Figura 9: Escenari SOAP.....	29
Figura 10: Entorn de desenvolupament VIM	30
Figura 11: Diagrama principal de seqüència de l'aplicació	31
Figura 12: Escenari inicial de l'aplicació	32
Figura 13: Representació del MVC.....	35
Figura 14: Representació de l'estructura de fitxers d'un mòdul Magento	37
Figura 15: Fitxers del mòdul Etailers SOAP	38
Figura 16: Exemple d'utilització de filtres	40
Figura 17: Estructura final del projecte i els seus fitxers	41
Figura 19: Exemple de llistat de comandes a Google Docs.....	46
Figura 20: Cel·la on el client pot demanar les comandes	47
Figura 26: Llistat de Google Docs	48
Figura 27: Exemple de llistat de vins d'un full de càlcul de vila viniteca.....	49
Figura 28: Producte configurable	49
Figura 29: Full de càlcul de Google Docs de youbuyoutdoor.....	50
Figura 30: Producte de motorstoreshop.....	50
Figura 31: Part del Google Docs de Motorstoreshop.....	51
Figura 32: Llistat de vendes i factures d'abonament	51
Figura 33: Part del document de llistat de productes de seleqto	52
Figura 34: Tasques de la planificació final.....	58
Figura 35: Diagrama de Grantt final.....	59
Figura 21: Relació entre el diccionari i el full de càlcul Google Docs	61

Figura 22: Backoffice de Magento	62
Figura 23: Website d'una configuració per defecte de Magento	62
Figura 24: Document Google Docs de productes.....	65
Figura 25: Error en un document de productes Google Docs.....	65

Capítol 1: Introducció

Cada cop que s'ha de fer una importació, l'empresa destina un programador que hi treballarà durant un dia o més. Aquest projecte pretén millorar aquest sistema i crear un programa que faciliti/millori la importació, disminuint, al seu temps, el cost.

Aquesta memòria tractarà sobre la millora del sistema d'importacions massives de productes a les botigues online creades amb la plataforma Magento [magento4]. L'actual sistema que té Magento suposa un cost molt alt de temps i treball per a l'empresa cada cop que es vol fer una càrrega de productes. Les importacions o actualitzacions massives es realitzen normalment quant es crea una botiga nova, per fer actualitzacions de preus o per actualitzar altres atributs a molts productes.

Aquest projecte es desenvoluparà a l'empresa The Etailers [etailers]. Aquesta empresa està dedicada a la creació i manteniment de botigues online, encarregant-se també de la publicitat i incentivar les compres en les seves botigues. Aquesta empresa és l'encarregada de incentivar aquest projecte i la principal beneficiada, invertint els recursos necessaris en el desenvolupament d'aquest projecte.

A més, si aquest projecte té èxit, es podria aportar a la comunitat de desenvolupadors de Magento, sempre amb l'aprovació de l'empresa.

En aquest document s'analitzaran els seus objectius, es farà un estudi de viabilitat i s'exposarà tot el seu desenvolupament

Motivació

El augment del ús d'Internet tant el l'entorn empresarial com en el l'entorn domèstic ha fet que el comerç electrònic s'hagi expandit d'una manera que mai ningú s'hagués imaginat, arribat a ser una acció quotidiana i habitual per molts usuaris d'Internet. Per part dels propietaris de les botigues físiques, Internet ha suposat trencar les fronteres dels mercats fent que, on abans una botiga només podia vendre al seu barri, ara pot arribar a vendre els seus productes a tot el mon. També ha suposat un apropament al client a l'hora de mostrar els productes, podent posar descripcions detallades, vídeos explicatius, possibilitat de fer preguntes, etc.

Per part del client, l'aparició del comerç electrònic ha suposat l'accés a un mercat molt més ampli ja que pot accedir a una gran quantitat de botigues que abans no tenia accés. També ha suposat una reducció dels preus dels productes i la possibilitat de fer una comanda al instant, sense haver de fer cues ni haver d'anar a la botiga física.

Per finalitzar, l'auge del comerç electrònic també ha fet possible la creació de molts llocs de treball, creant empreses que es dediquen exclusivament a la venda online, a la creació de botigues online o a la publicitat online. Per aprofundir més en el món global del comerç electrònic, es recomana els articles [article] i [article2] de la bibliografia.

Objectius del projecte

L'objectiu principal d'aquest projecte consisteix en millorar l'actual sistema d'importació de productes que té Magento. Per tal d'assolir aquest objectiu, s'han marcat els següents objectius:

- Simplificar el sistema d'importació de productes, fent que molts dels atributs que necessita Magento per importar un producte s'emplenin automàticament.
- Usar de l'API de Google Docs per, en lloc d'introduir els productes des d'un csv, s'introdueixin des d'un full de càlcul Google docs.
- Poder importar els productes des d'un full de càlcul Google Docs a una botiga Magento.
- Importar diferents productes en diferents idiomes.
- Importar imatges dels productes introduïts al Google Docs.
- Identificar i actualitzar els productes que ja existeixen a la botiga on-line amb el contingut que s'ha introduït a Google.
- Validar els valors que introdueix el client a Google.
- Estudiar el llenguatge Python i l'API de Google docs per a la realització de la importació.
- Estudiar la compatibilitat entre PHP i Python per a la interacció entre Magento i el programa.
- Realitzar un parser per poder fer un estudi de mercat de tots els productes que hi hagi al Google Docs, fent una comparativa de preus.

Justificació dels objectius

Actualment, per importar un producte, es necessita uns 70 atributs que s'han d'introduir en un full de càlcul tipus csv. Molts d'aquests atributs sempre tenen el mateix valor però per a un usuari no expert, el fet de gestionar tants atributs es pot fer difícil i fa que la possibilitat de cometre errors sigui alta. Per tant, és necessari reduir aquests atributs al mínim possible.

La idea de la utilització de l'API de Google és degut a que es vol substituir el format csv per fulls de càlcul de Google a l'hora d'introduir els productes. Aquesta substitució ve justificada per aquests motius:

- Els clients podran manipular el full de càlcul dels productes des de qualsevol ordinador amb connexió a Internet.
- Es disposarà d'un historial de totes les modificacions al document. Aquest historial estarà implementat per part de Google i permetrà veure, detectar, i corregir possibles

errors d'introducció d'informació. A més, si és necessari, permet la recuperació de dades que s'han eliminat per error.

- Permetrà treballar, al mateix temps, diferents persones en el mateix document fent més fàcil el treball en grup per part dels clients. A més, Google mostra el treball de cada usuari a temps real i es pot fer un seguiment en directe del que fa/n el/els usuari/s.
- Els fulls de càlcul de Google Docs tenen la gran majoria de funcions d'un full de càlcul Microsoft Office. A priori, l'aprenentatge i la manipulació per part dels clients del document serà fàcil i amigable.
- S'Aprofitarà l'interfície gràfica ideada per Google per a la manipulació del document.
- Possibilitat de descarregar-se el document en altres formats per major comoditat de l'usuari i, un cop manipulat, poder importar el document al mateix full. Això també pot ser útil per als clients que no disposin o no vulguin treballar amb Google docs, donant el document en format Excel i importar-lo a Google després.

El resultat final, mitjançant aquests fulls de càlcul de Google docs, és que el client haurà d'omplir tots els seus productes amb el seu nom, número d'identificació (d'ara en endavant: sku), descripció, etc. Un cop tots els productes siguin a Google docs, el programa haurà d'agafar aquest full de càlcul i importar tots els productes a la botiga. Cas que el producte ja existeixi, el programa haurà d'actualitzar el seu contingut amb el que s'ha introduït a Google. Això permetrà poder fer tant importacions com actualitzacions de productes (per exemple, actualitzacions massives de preus).

Per a què un parser de productes?

Una de les claus del comerç electrònic és que els preus es trobin sempre a nivell de mercat. Sempre pot passar que surti una web amb millors preus que els que la pròpia botiga ofereix i les vendes baixin sense motiu aparent.

Per poder controlar els preus de les altres botigues, es realitzarà una altra millora que consistirà en, un cop emplenat el Google doc amb tots els productes, es farà un estudi de preus dels propis productes amb els de les altres. Les webs on es compararan els preus seran donades pels clients i seran ells qui controlaran els preus dels seus competidors.

Per què Python i PHP?

El programa es realitzarà, a priori, amb el llenguatge de programació Python. L'elecció d'aquest llenguatge és basa en diferents motius:

- **Auto aprenentatge:** Coneixement d'un nou llenguatge de programació.
- **Foment d'aquest llenguatge per part de l'empresa:** l'empresa té algunes webs o programes realitzats amb aquest llenguatge (ex: www.sorteisimos.com, desenvolupat amb el Framework Django). Si més programadors coneixen aquest llenguatge, serà més fàcil el seu manteniment i la seva millora. També ajuda a la implementació més ràpida de noves plataformes amb Python.

- **Python i Google:** Python és un dels llenguatges de programació més utilitzats per Google. L'API de Google Docs està més actualitzada per Python (actualment versió 3) que no la desenvolupada amb PHP (actualment està a la versió 1).

Tot el que fa referència a l'apartat de Magento es realitzarà amb PHP. Magento està programat amb PHP i tot el que sigui necessari per interactuar amb ell, es farà amb aquest llenguatge. Ja hi ha programada una llibreria d'interacció SOAP amb Magento feta amb PHP que actualment es fa servir per realitzar actualitzacions de productes remotament.

El programador encarregat d'aquest projecte ja té experiència en PHP i amb l'API de SOAP de Magento, fet que facilitarà la creació i la manipulació de dades dels productes.

Prioritat dels objectius

A continuació es prioritzaran els objectius i es justificarà el perquè s'ha donat importància a segons quins objectius:

(ordenats de major prioritat a menor prioritat)

1. **Estudiar el llenguatge Python i l'API de Google:** Es considera un objectiu necessari i indiscutible per poder començar el projecte. Com part del projecte es realitzarà amb aquest llenguatge i el programador no té coneixements de Python, s'ha d'estudiar com funciona i la compatibilitat amb el llenguatge PHP. Aquest objectiu es preveu realitzar-lo a mesura de que es vagi programant el programa i amb la lectura d'alguns manuals [python].
2. **Simplificar el sistema d'importació, fent servir Google Docs, d'importar productes:** Aquests és l'objectiu més important del projecte (tot i que prèviament és necessari fer l'estudi abans de començar. Per aquests motius s'ha introduït en segon lloc). Si no s'assoleix aquest objectiu, el projecte no tindrà utilitat i no es podran aportar millores a l'actual sistema d'importació.
3. **Estudiar la compatibilitat entre Python i Magento:** A priori, tot sembla indicar que els programes es poden realitzar amb les dues tecnologies sense cap problema però es pot trobar alguna situació que faci que els dos llenguatges entrin en conflicte. És important veure si tot el que es va realitzant té algun problema d'interacció amb l'altra part del projecte. Com que el programador assignat al projecte no té coneixements dels problemes que suposa la implementació de les dues plataformes, es posa en tercer lloc i l'estudi es realitzarà durant el desenvolupament ja que l'objectiu és tenir un prototip funcional el més aviat possible i, si s'ha de reescriure tot el codi en un altre llenguatge, l'empresa considera que tota la feina desenvolupada en Python no s'haurà de repetir i serà ràpid de reformular. Tampoc es considera tant prioritari com les tasques 1 i 2 ja que l'empresa vol que el programador realitzi un estudi de Python i, sobretot, que s'acabi el projecte per a la seva utilització immediata.
4. **Importar les imatges dels productes:** Si s'aconsegueixen les tasques 1, 2 i 3, el projecte ja serà operatiu però es consideraria ineficaç. Un sistema d'importació sense

un sistema per associar els productes que estan a Google Docs amb les seves imatges no es podria considerar com un projecte acabat. No es recomanable publicar un producte si no té una imatge que el pugui representar. Es considera una tasca important degut a que l'objectiu principal de tot e-commerce, es vendre el producte i millorar la seva presentació davant la competència.

5. **Actualitzar productes:** l'actualització massiva de productes és una part molt útil en la gestió d'un e-commerce però l'empresa ja té més experiència i més eines d'actualitzacions que d'importació. Actualment l'empresa realitza scripts SOAP "a mida" per fer algunes actualitzacions de preus o d'altres atributs. Es dóna prioritat a la realització d'un actualitzador més elaborat i més genèric però primer s'han d'assolir les tasques 1, 2, 3 i 4.
6. **Importar diferents idiomes:** Magento té la capacitat de tenir un producte amb més d'un idioma. L'empresa considera molt útil que aquest programa també permeti la importació i/o l'actualització dels productes en diferents idiomes. Aquest objectiu és important però està considerat de baixa prioritat ja que la majoria de webs es centren en un sol idioma.
7. **Realitzar un comparador de preus:** La realització d'un parser per comparar preus de la competència és una eina molt útil però és difícil fer un parser genèric per a totes les webs sol·licitades. És molt possible que s'hagin de fer parsers a mida per algunes determinades webs i pot endarrerir el projecte. L'empresa disposa d'un equip de màrqueting que seran els principals beneficiaris d'aquest projecte.

Equip del projecte

A continuació es farà un llistat de les persones encarregades del desenvolupament del projecte així com de les responsabilitats que assumeixen.

Nom	Responsabilitat	Descripció
Greg Robbins	Cap de projecte Etailers	Defineix, gestiona, aconsella, planifica i guia el projecte.
Ramón Martí	Director del projecte UAB	Supervisa la feina de l'estudiant
Oriol Augé	Dissenyador	Disseny i desenvolupa tot el que fa referència al projecte tant a nivell de codi com d'estructura
	Analista	Col·labora amb el cap de projectes en l'estudi de viabilitat i la planificació. Analitza l'aplicació: arquitectura, metodologia, especificacions, estàndards... Participa en el disseny i la validació.
	Programador	Desenvolupa el projecte d'acord amb el disseny previst.
	Tècnic de proves	Encarregat dels tests corresponents i detectar qualsevol mal funcionament.

Què es vol aconseguir?

El que es vol aconseguir és simplificar, reduir el percentatge d'error, reduir el temps dedicat a les importacions i afegir noves funcions. A part, el programa hauria de ser capaç de validar les dades fent que es pugui localitzar fàcilment els errors que el client produeixi i corregir-los automàticament.

Estat de l'art

En aquest apartat es descriuen les plataformes de comerç electrònic (e-commerce) per poder analitzar les actuals solucions open source que hi ha actualment al mercat i poder veure, amb més claredat, perquè s'ha escollit Magento en lloc de les altres plataformes.

Per acabar, un cop analitzades les plataformes indicarem els motius principals de perquè considerem oportú millorar el sistema actual d'importació de Magento.

Plataformes de comerç electrònic (e-commerce)

Actualment hi ha varies plataformes per crear una botiga online. Els motius entre escollir una plataforma o una altra poden ser el cost, la rapidesa de l'aplicació, la simplicitat de treball, etc. Si ens centrem en les de codi lliure, les més famoses són Magento, OsCommerce [oscommerce], Joomla [joomla][joomla2](amb plugins per crear un e-commerce), Drupal [drupal][drupal2](també amb plugins) i Prestashop [prestashop] entre d'altres.

OsCommerce

Aquesta plataforma va néixer el any 2000, consisteix en una plataforma de codi obert que et permet instal·lar i gestionar una botiga online. Conté les dos parts bàsiques com la part visible, on es poden visualitzar els productes, i una zona d'administració on es pot gestionar tota la informació referent a la botiga. Està programada amb PHP i MySQL com a servidor de base de dades. OsCommerce té totes les funcions bàsiques que es necessiten per poder gestionar una botiga, com ara crear/actualitzar productes, permetre diferents opcions de pagament, diversitat en la configuració dels enviaments, etc.

Aquesta plataforma va ser la més important a nivell open source fa un anys però ara ha quedat desfasada pels seus competidors. També ha perdut comunitat de desenvolupadors que han anat a Magento o a Prestashop. Un dels grans avantatges respecte Magento és la seva base de dades. Té molt poques taules en comparació amb Magento i la seva línia d'aprenentatge és molt millor que Magento. Tot i això, es va optar per Magento degut a que OsCommerce no té tantes funcionalitats implementades a nivell de backoffice i està perdent suport. Per l'empresa escollir Magento en lloc de OsCommerce permet als programadors especialitzar-se en desenvolupar peticions específiques per a les botigues en lloc de programar funcions bàsiques que Magento ja té de sèrie i no OsCommerce. També assegura tenir actualitzacions de la plataforma amb més regularitat que OsCommerce, major suport amb mòduls i disposar d'una major comunitat.

Joomla i Drupal

Joomla i Drupal són dos gestors de continguts especialitzats en la creació de webs. La seva gran característica és la gestió de contingut web d'una manera molt senzilla, guardant i modificant el contingut en una base de dades enlloc de fitxers físics. Això facilita la feina a una gran quantitat d'usuaris que, sense tenir grans coneixements de web o html, poden modificar el contingut de la seva web amb molta facilitat i, a més, amb la possibilitat d'instal·lar plugins per afegir funcionalitats extres. Els CMS Joomla i Drupal estan programats amb PHP i funcionen sobre MySQL.

Com s'ha analitzat anteriorment, aquestes dues plataformes no són e-commerce en si. Són gestors CMS que, amb la instal·lació de plugins específics, poden fer les funcions d'un e-commerce. Aquí està el seu gran problema. Si es vol tenir una botiga bàsica i senzilla, aquests dos gestors de CMS compleixen perfectament els requisits però, al tenir un negoci encarat a la creació i al manteniment de botigues on-line, no és recomanable fer servir gestors de CMS i dependre de plugins per fer la funció d'una botiga. Per altra banda el seu punt fort enfront de Magento és la seva gestió dels CMS, molt millor i més suportada que a Magento, on la gestió de CMS és bàsica.

Prestashop

Prestashop consisteix en una altra plataforma e-commerce open source. Els seus inicis van ser a França però ja està completament en anglès i disposa de les seves respectives traduccions. Prestashop conté les funcionalitats bàsiques que necessita tot e-commerce per funcionar i, a més, té una integració molt bona amb AJAX, sent de les tres plataformes e-commerce la que més disposa d'aquesta tecnologia, donant un punt de innovació davant les altres. A més aquesta plataforma ha estat premiada en 2010 i 2011 com la millor aplicació de comerç electrònic.

Prestashop està creixent a un gran nivell i tot apunta a que pot superar Magento en un futur, però encara no té tantes funcionalitats ni tant suport amb mòduls com Magento. Quan Prestashop hagi arribat al nivell de Magento a nivell de funcionalitats, es podrà estudiar la possibilitat de crear botigues amb aquesta plataforma però ara mateix encara no es contempla aquesta opció.

Magento

Magento és la tercera plataforma de comerç electrònic i en la que es basa aquest projecte. Consisteix en una plataforma open source que va començar el 2007 i que s'ha anat ampliant fins a ser una de les plataformes més utilitzades e-commerce del moment. Magento té uns dels backoffice més complerts i està programat completament amb PHP i, a més, funciona sobre MySQL.

Tot i ser una plataforma open source, Magento [magento3] està suportat per l'empresa Varien que dóna suport al projecte i que està al darrere de tot el desenvolupament. A més, fa poc Magento ha estat comprat per Ebay. Això li dóna més garanties i fa pensar que

Magento estarà en constant creixement. Aquest suport no està contemplat en altres plataformes open source com, per exemple, OsCommerce.

Magento té una gran comunitat i en constant creixement. Cada dia es creen nous mòduls (gratuïts o no). A més, Magento està construït amb Zend Framework. Zend Framework és un sistema de desenvolupament d'aplicacions webs fet amb PHP5 i completament orientat a objectes. Això fa que el codi sigui molt més modular i més actual que altres plataformes (com per exemple, OsCommerce). A part, tots els mòduls creats amb aquesta tecnologia acostumen a tenir un cost menor de programació degut a que moltes funcionalitats ja estan implementades en el Framework. Això fa que el cost dels mòduls privats de Magento siguin menors que els d'altres plataformes.

Magento està programat íntegrament amb PHP. Aquest llenguatge és un dels més coneguts a nivell d'Internet. Quant es busca programadors especialitzats en aquest llenguatge, pot ser més fàcil que si es busquen programadors especialitzats amb altres llenguatges de programació.

Un altre punt a favor de Magento és tot el seu backoffice. Magento té un dels backoffice més forts del mercat de plataformes e-commerce de codi lliure. Magento permet gestionar moltes opcions i configuracions internes d'una botiga sense la necessitat d'instal·lar cap plugin. Altres plataformes (com Drupal) permeten una gran gestió a nivell de CSM que no té Magento com hem analitzat però no podrà tenir tantes funcionalitats de gestió com Magento inicialment.

Per què millorar el sistema d'importació de Magento?

L'actual sistema d'importació de Magento fa ús dels fitxers amb format csv per introduir gran quantitat de productes i poder importar-los. Cal dir que cada producte s'ha d'introduir amb més de 70 atributs que molts dels quals són de configuració del producte dintre la botiga Magento.

Per a un client no expert, la importació de productes pot ser un procés difícil i té el risc de cometre una gran quantitat d'errors. Per simplificar aquest sistema, en l'empresa Etailers es demana al client que emplei únicament uns atributs essencials del producte (descripcions, skus, preus, etc) per facilitar-li la feina. Un cop el client ha omplert el csv amb tots els productes corresponents i els seus atributs, l'empresa, manualment o amb l'ajuda de macros, emplena tots els atributs restants per fer una importació inicial en un entorn de proves i, un cop analitzat que els productes s'han importat correctament, es fa la importació final.

Aquest sistema té algunes mancances i aspectes que es poden millorar com:

- Al ser un procés manual, aquest mètode és molt lent i amb una gran probabilitat d'errors humans. El temps mínim estimat en fer una importació (entre correccions d'errors i tests) és d'un dia i mig.
- Un altre problema són els fitxers csv. El client els envia per email o per un altre sistema el seu csv on hi cap una probabilitat que hagi comés un error. Quan es modifiquen els csv, també hi ha la probabilitat de cometre equivocacions en omplir el fitxer. Si sumem

tots els factors de risc tenim massa possibles errors i es perden moltes hores en detectar-los i corregir-los. A més, cal sumar la quantitat de fitxers csv que es creen quant es van modificant (sigui per part del client o de l'empresa). Tot i no ser un factor rellevant, la gran quantitat de versions del document pot comportar confusió i dificulta les tasques d'importació.

Estudi de viabilitat del projecte

Els punts que s'haurien de tenir en compte a l'hora de realitzar el projecte són:

- Ha de ser un sistema molt fàcil d'utilitzar per part dels clients. La seva corba d'aprenentatge ha de ser molt curta, ràpida i, per als que ja estaven acostumats a l'antic sistema, no els ha de suposar gaire esforç passar al nou sistema.
- Ha de ser un projecte escalable, amb capacitat per aplicar-se a una gran diversitat de botigues amb característiques diferents.
- Ha de ser un sistema amb un manteniment mínim per part dels treballadors.
- El sistema ha de reduir notablement les hores dedicades a totes les importacions.
- Un possible problema pot ser pujar les diferents fotos dels productes a la web. Això pot requerir fer ús d'algun programa extern ja que pot ser molt complicat importar les imatges de cada producte a Google docs. Aquest programa serà realitzat des de zero i també s'inclourà a la planificació temporal.
- S'ha de trobar dins dels paràmetres legals.
- Cost econòmic baix.

Viabilitat legal

En el full de càlcul que s'utilitzarà per realitzar aquest projecte es poden trobar informacions de caràcter personal o comprometedores per a una empresa. D'acord amb la Llei orgànica 15/1999 de protecció de dades de caràcter personal [Llei] hem de tenir en compte que, segons l'article 9 de seguretat de les dades i l'article 10 del deure del secret professional, es garantiran totes les restriccions i mètodes per protegir les dades que l'empresa introdueixi en el document. Les restriccions seran les que ofereix Google docs a l'hora de compartir un document que, inicialment, estarà limitat als treballadors de l'empresa i als treballadors del client. Els usuaris que tindran accés aquest document sempre estarà acordat prèviament entre empresa i client.

Per el que fa a la Llei LSSI (Llei de serveis de la societat de la informació) [Llei2] afecta en aquest projecte en el moment en que s'ha de garantir que els productes han de tenir establert el tipus d'IVA que se'ls hi aplica. Abans de llançar una web online, l'empresa s'encarrega del compliment d'aquesta Llei (comprovant si es mostra el CIF de l'empresa, telèfon, correu electrònic, codis de conducta per comprar, etc.) però s'haurà de garantir que el programa pugui importar l'IVA de cada producte. Tot i això, cas que algun client estigui en contra de publicar les dades a Google docs, faríem servir el procediment antic d'importació per tal d'evitar conflictes.

Pel que fa al parser, tota informació extreta amb ell serà pública i accessible des de la web, fet que garanteix que no s'agafarà informació de caire personal de les webs on es farà l'estudi. La informació serà la mateixa que podria obtenir un usuari normal navegant per la web.

Viabilitat tècnica i operativa

Pel que fa a l'apartat de hardware/software, l'empresa disposa dels següents recursos:

Servidors de Producció:

- Un servidor Linux amb el sistema operatiu Debian, especialitzat amb el tractament de PHP. Té instal·lat el mòdul de PHP i disposa d'una instal·lació Apache i Nginx. Aquest servidor és l'encarregat de servir tota la part visible de la botiga. D'ara en endavant, aquest servidor s'anomenarà "frontend". També disposa de llibreries Python per l'execució d'una web feta amb aquest llenguatge.
- Un servidor Linux amb el sistema operatiu Debian especialitzat en el tractament de base de dades. El mòdul instal·lat és MySQL i és l'encarregat de servir totes les peticions MySQL que requereix el servidor frontend. D'ara en endavant, aquest servidor s'anomenarà "dumper".

Servidors de Pre-producció

- Un servidor Linux amb el sistema operatiu Debian amb instal·lacions de PHP, Apache, Nginx, MySQL i tot el necessari per gestionar una botiga Magento.

El projecte es programarà amb els llenguatges PHP i Python perquè, el *core* de Magento està programat íntegrament amb PHP i Python serà usat per interactuar amb l'API de Google. A més, permetrà conèixer un nou llenguatge de programació que, com s'ha dit anteriorment, és un dels objectius del projecte.

Si més endavant es necessita alguna configuració nova que no està contemplada en els servidors, l'empresa ha acceptat instal·lar el software necessari per fer funcionar el programa. Un exemple pot ser l'API de Google o noves llibreries de Python. Pel que fa a la part de hardware, creiem a priori que es compleixen els requeriments necessaris per a que un programa funcioni sense problemes.

Viabilitat econòmica

A l'utilitzar tecnologia gratuïta (tant Google docs com Magento són gratuïts), aquest projecte només costarà les hores de treball del treballador. Aquest cost ja ha estat contemplat per l'empresa i s'ha considerat com acceptable.

Conclusió

Com a conclusió final, es pot considerar que aquest projecte és viable descartant el problema de les imatges ja previst en la planificació.

Planificació temporal del treball

Aquest projecte consta d'una planificació inicial però, per diferents motius que s'explicaran en aquest apartat, la seva planificació ha tingut varis canvis no previstos que han fet variar la planificació al llarg del projecte. Per poder entendre i poder explicar els diferents canvis de planificació, explicarem la política de prioritats de l'empresa per gestionar els diferents projectes:

1. Errors crítics en les e-shop.

Els errors crítics de les e-shops són errors que impedeixen les vendes a la botiga on-line. Un error crític pot ser que es detecti un mal funcionament a la passarel·la de pagament, no poder afegir productes a la cistella de la compra, no poder entrar a la pàgina web degut al servidor, etc. Quant un client o una persona de l'equip tècnic detecta aquest tipus d'errors s'ha de prioritzar la seva solució assignant els recursos que el cap de desenvolupament vegi convenient. Si es detecta que la seva solució no és immediata, les solucions de l'empresa poden ser assignar més recursos al problema o deshabilitar l'opció (si és possible) de la botiga per poder continuar operant sense problemes en la e-shop però internament es continua treballant fins a trobar una solució.

Això pot fer que el programador hagi de deixar el projecte en més d'una ocasió per poder solucionar els problemes que hi hagin en les e-shops.

2. Creació de noves e-shops.

L'actual model de negoci de l'empresa es basa en ingressos per el manteniment de les webs i un percentatge que s'emporta l'empresa de cada venda. Si analitzem aquest model, podem veure que a l'empresa l'interessa que es prioritzi la creació de noves e-shops ja que, així, obté més beneficis i més ingressos per poder-los invertir en altres projectes o material. És prioritari el llançament dels e-commerce perquè, un cop s'ha llançat a producció, el client comença a pagar el manteniment de la web i, a més, comença a rebre el percentatge de beneficis per cada venda.

Què pot comportar això al nostre projecte? Si hi ha alguna web important que ha de sortir a producció o hi ha algun projecte que necessita més recursos dels que té assignats, el programador deixarà el projecte per centrar-se en aquesta tasca.

3. Manteniment de les e-shops actuals.

El manteniment de les e-shops es pot centrar tant en actualitzar Magento com en realitzar peticions que demana el client. Pot ser el client demani una nova maquetació o una funcionalitat extra que la web no tenia i s'hagi de desenvolupar de nou. Quan arriben aquestes peticions, l'empresa valora la seva prioritat davant d'altres clients que també han demanat funcionalitats extres. Un cop s'han valorat, s'estima un temps de desenvolupament, un (o varis) desenvolupadors i s'informa al client de quan tindrà o es començarà a desenvolupar aquesta funcionalitat.

4. Creació/manteniment dels projectes interns.

Els projectes interns són programes que s'utilitzaran per millorar o facilitar algunes de les feines que han de desenvolupar els clients o els empleats. També pot ser una botiga que és de la pròpia empresa i no té cap client darrera (www.motorstoreshop.com). Quan entra un projecte, es fa una valoració i una anàlisi del que aportarà cada projecte i si el cost de desenvolupament és inferior als beneficis que se n'obtindrà. Si l'empresa veu convenient crear el projecte, s'assignen els recursos necessaris i es comença a desenvolupar el projecte sempre i quan no hi hagi cap altra tasca més prioritària.

Aquest projecte està considerat com a projecte intern. Això fa que si hi ha alguna tasca que sigui de major prioritat que aquest projecte, es deixarà de treballar temporalment en ell i s'assignaran els recursos a les noves tasques.

Les hores d'oficina de l'equip són de 5 hores diàries. D'aquestes 5 hores, entre 2 i 3 hores de treball estan previstes a aquest projecte.

A les diferents planificacions també s'ha previst el problema d'importar les imatges dels productes. Si al final no és necessari i es pot resoldre fàcilment, aquest temps i totes les tasques associades s'eliminaran i el temps restant s'aprofitarà per avançar.

Planificació

Les tasques a assolir són les següents (figura 1):

El cost per assolir aquestes tasques està calculat per la següent manera:

- Cost per la llicència i utilització de Magento: 0 €
- Cost per la llicència i utilització de Google Docs: 0 €
- Hores de treball aplicades al projecte: s'ha estimat que s'assignaran al voltant de 350 h de treball al projecte. $350 \text{ h} \times 4 \text{ €/h} = \mathbf{1400 \text{ € de cost al projecte per part de l'empresa.}}$

El cost s'ha calculat en base al sou del treballador que s'ha assignat al projecte. Al ser un contracte de pràctiques, el sou del treballador és reduït ja que no està com a treballador d'empresa. Aquesta regla s'aplica a tots els càlculs de costos de tasques.

Task	Effort	Unique ID
▼ 1) Informe previ	2w 4d 1h	8
• 1.1) Definició d'objectius	0,5h	2
• 1.2) Definició de les parts	0,5h	9
• 1.3) Redactat del document	1w 3d	10
• 1.4) Revisió i supervisió	1w 1d	11
▼ 2) Anàlisis	6w	73
• 2.1) Anàlisis del problema	2w	74
• 2.2) Anàlisis del importador de Google	2w	75
• 2.3) Anàlisis del comparador	2w	76
• 3) Disseny de classes	1w	79
▼ 4) Desenvolupament	19w 3d	14
▼ 4.1) Google Docs	14w 1d	16
• 4.1.1) Estudi de python i api de google docs	1w	15
• 4.1.2) Connexió i obtenció de documents google docs	1w	17
• 4.1.3) Manipulació del document	2w	18
▼ 4.1.4) Enviament dels productes a la web	5w 1d	71
• 4.1.4.1) Creació dels productes en la e-shop	2w	64
• 4.1.4.2) Actualització dels productes en la e-shop	4d	66
• 4.1.4.3) Actualització dels productes en diferents idiomes	4d	68
• 4.1.4.4) Validació de camps i correcció d'errors	1w 3d	19
• 4.1.5) Sistema de log	2w	21
• 4.1.6) Test	2d	22
▼ 4.1.7) Importació d'imatges	2w 3d	24
• 4.1.7.1) Estudi de com importar les imatges. <i>A pactar amb l'empresa el disseny</i>	1d	25
• 4.1.7.2) Implementació	2w	27
• 4.1.7.3) Test	2d	26
▼ 4.2) Estudi de la competència	5w 2d	28
• 4.2.1) Sol·licitar diferents webs per fer l'estudi <i>En cas de no rebre cap web, s'agafarà 3 d'exemple pactades amb l'empresa</i>	3d	29
• 4.2.2) Estudi dels webs obtinguts	2d	30
• 4.2.3) Implementació d'un parse	2w	31
• 4.2.4) Mostrar els resultats en google docs	2w	32
• 4.2.5) Test	2d	33
• 5) Realització de la memòria	20w 1h	43

Figura 1: Tasques planificació 1

Descripció dels principals grups de tasques

Millora de la importació amb Google docs (tasques 4.1):

Aquest grup de tasques tracta tot el que fa referència a la connexió, manipulació i validació dels camps a introduir al full de càlcul. També s'han reservat dues setmanes i tres dies al possible problema de les imatges ja que és molt possible que s'hagi de fer un programa independent per solucionar aquest problema.

Estudi de mercat de les diferents webs de la competència (tasques 4.2):

Ara mateix ja s'ha parlat amb l'empresa sobre com fer aquests estudi i apreciem que s'haurà de fer un estudi específic per cada web (estudiar l'estructura i crear un parser per a cadascuna). En cas que trobem un mètode més òptim es passarà a fer l'estudi amb una altra plataforma que no sigui un crear el parser.

Tot i ser un programa molt "individual" i "concret" per a cada web, veiem necessari fer-ne ja que aportarà una informació molt valuosa del mercat a Internet.

Podem veure el diagrama de grantt en la figura 2:

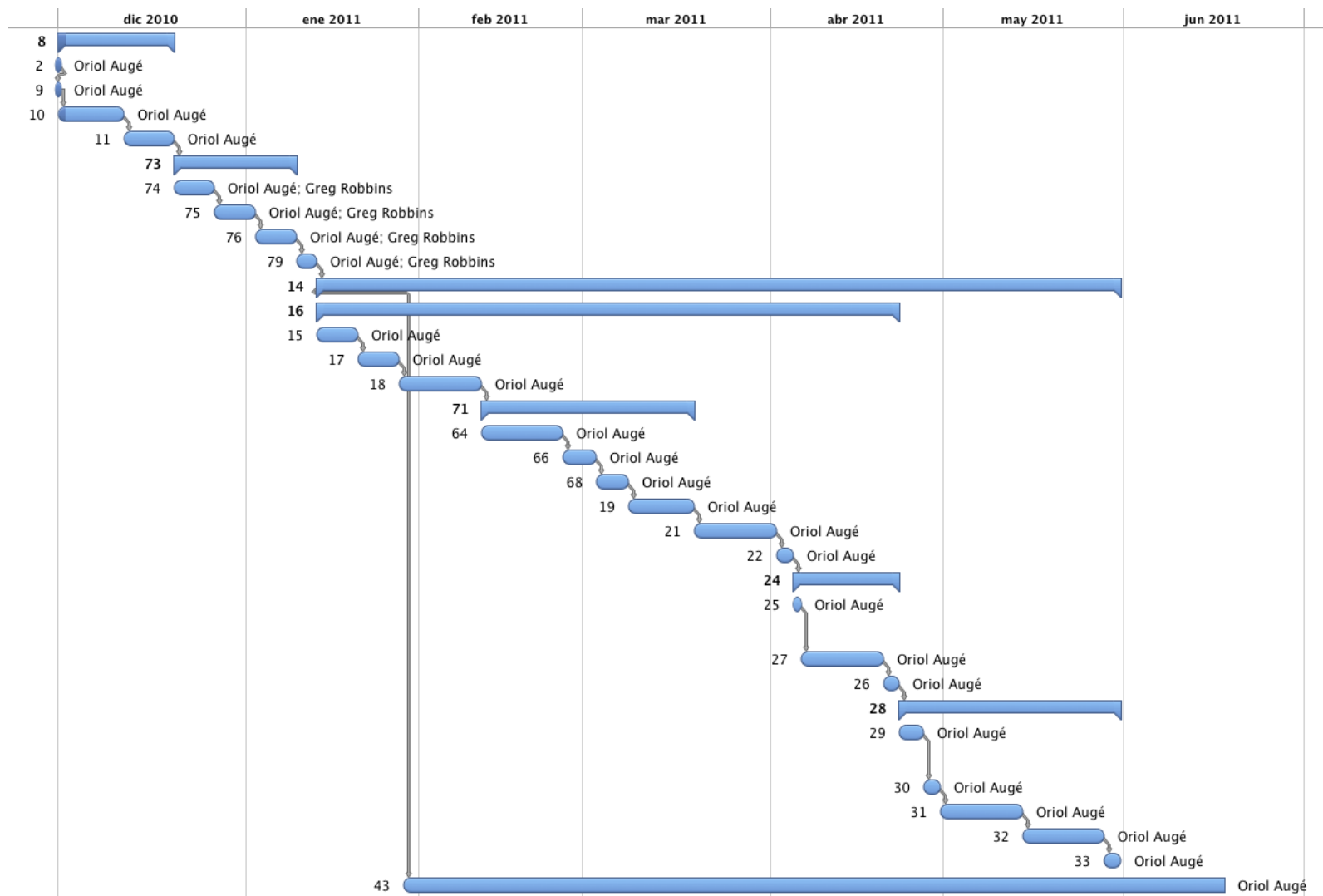


Figura 2: Diagrama de grantt. Cada tasca està identificada amb la seva Id.

Canvis en la planificació

Aquesta planificació inicial no es va poder assolir degut a que l'empresa va veure necessari destinar els recursos d'aquest projecte a altres amb més prioritat. Això va alentir els temps de finalització i inici de les tasques. Les tasques més perjudicades van ser l'anàlisi i el disseny de classes que es van endarrerir i el desenvolupament que no es va poder començar en la data prevista.

A més, durant l'anàlisi del projecte, es va començar arribar a la conclusió de que no era tant prioritari fer un comparador de preus. Aquesta conclusió ve de les diferents raons que s'explicaran a continuació:

1. Plantejament d'un nou model de negoci.
2. Previsió de gran temps en manteniment.
3. Poca implicació dels clients.
4. Introducció de noves eines.

Plantejament d'un nou model de negoci

Magento suporta la gestió de diferents botigues amb la mateixa base de dades. Cada botiga consta de la seva vista de botiga i els seus productes. La gestió de les diferents botigues està pensada per a un sol client, fent que ell pugui veure des de el seu backoffice tota la informació de tots els productes. Per posar un exemple, un client pot gestionar la seva botiga principal, un outlet i una marca blanca des del mateix backoffice (figura 3):

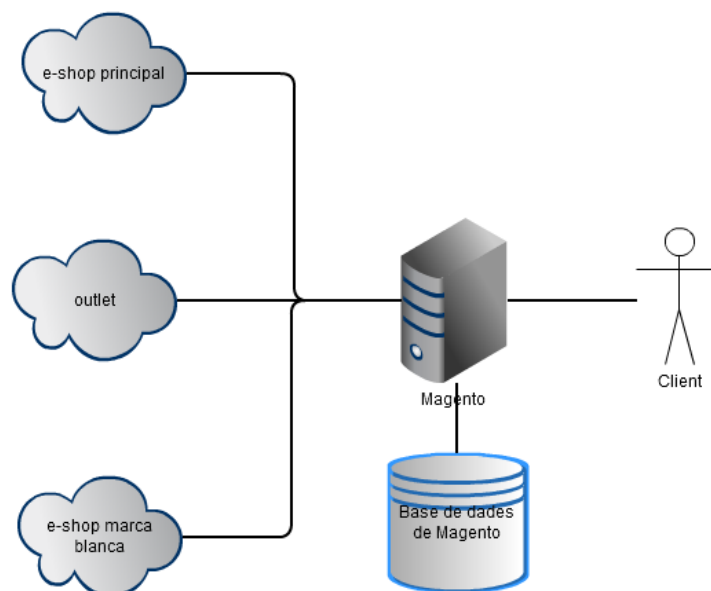


Figura 3: Escenari 1 d'un client amb diferents e-shops i un sol backoffice

El problema per l'empresa ve generat quan, des d'una sola instal·lació de Magento, es vol crear diferents botigues per a diferents clients. Actualment tots els clients podrien veure els productes que té l'altre client i això, podria generar que algun client borrés, per error, productes d'un altre client o veies informació confidencial d'altres clients.

Donat que el programa que es vol crear pot importar i actualitzar productes, la gestió de la botiga es podria fer des del mateix google docs. Això faria que cada client només tingués accés a un full de càlcul google docs, on constarien tots els seus productes i podria gestionar-los sense veure els productes dels altres clients (figura 4):

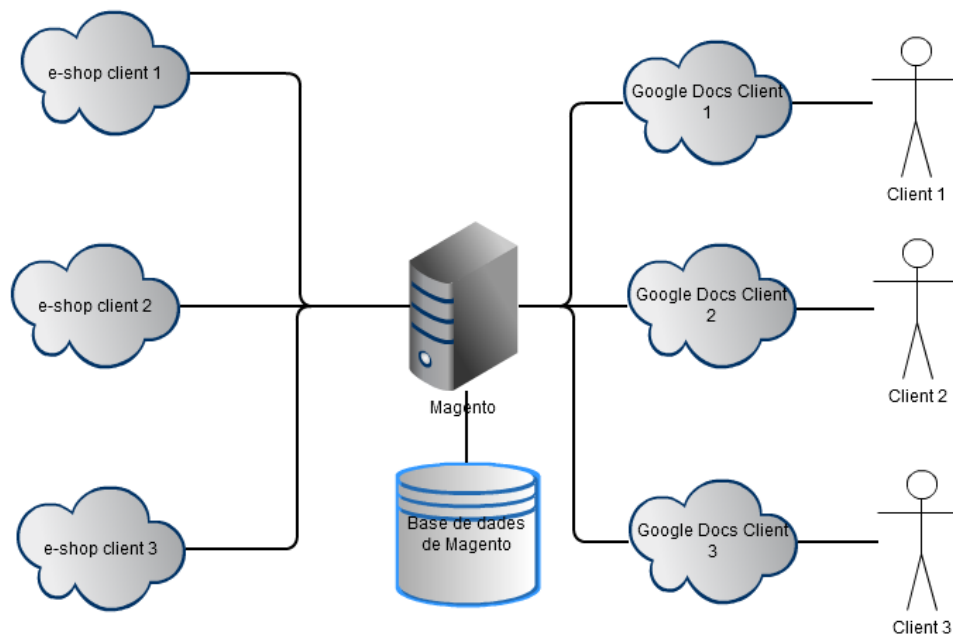


Figura 4: Escenari 2 amb varies clients treballant amb les seves botigues a través de Google Docs

Com es pot veure en la figura 4, cap client pot veure i accedir als productes de l'altre client ja que ells no accedeixen al backoffice de Magento degut a que ells treballen directament a Google Docs però, en realitat, tots treballen sobre la mateixa base de dades i instal·lació de Magento. A més, el client no sabrà que està compartint la base de dades amb un altre client.

Aquest model de negoci està pensat per a clients amb pocs productes. Per l'empresa això permet un manteniment ràpid i fàcil ja que si hi ha un error, millora o actualització que s'ha de corregir, es corregirà en tots els clients. El cost també està pensat que sigui baix i que les plantilles de les botigues siguin molt estàndards. A més, només els treballadors d'Etailers podran accedir al backoffice de Magento i, des de allà, la gestió de tots els productes de tots els clients també pot ser molt ràpida i fàcil.

Aquest model de negoci aporta nous reptes. Com pot el client gestionar les seves vendes? Com pot gestionar l'estoc dels productes? Aquestes incògnites s'han plantejat fent nous informes de google docs que aportin tota aquesta informació i, per aquest motiu s'ha exclòs el comparador de preus. Ara, s'ha introduït el informe de vendes i l'informe d'estoc/actualitzador d'estoc.

Previsió de gran temps de manteniment

El comparador de preus és una eina útil però que requereix força temps d'implementació i, el gran problema, és que no és una eina "genèrica". El comparador en la seva essència, és un parser de webs on agafarà la informació dels productes i els compararà. Si algun dia les webs canvien el format o fan algun canvi que afecta a l'estructura el parser fallarà i, amb això, totes les seves estadístiques. Això fa preveure un manteniment constant que l'empresa, ara mateix, no disposa i es prefereix assignar els recursos a altres problemes o projectes de més prioritat.

Poca implicació dels clients

Un altre problema que es veu al comparador és que han de ser els mateixos clients qui aportin les webs que venen els mateixos productes i que també siguin ells qui es preocupin de fer l'estudi. S'ha vist en experiències passades que demanar aquestes webs al client és un procés lent i, moltes vegades, el client ha donat webs incorrectes o, al fer el recorregut de les webs de la competència, ell mateix ja ha vist com està la seva e-shop en el mercat i ja no ha calgut fer tal estudi.

Introducció de noves eines

Cada cop s'utilitzen més els comparadors d'empreses externes que, tot i que al inici d'aquest projecte no estaven contractats, ara ja s'estan utilitzant per l'equip de Marketing online. Aquests comparadors ja aporten la informació que el nostre parser volia aconseguir i, com que és vital per l'equip de Marketing fer sortir les webs en els comparadors, ja s'aprofita aquesta eina contractada per fer l'estudi de la competència.

A més a més, el mateix equip de màrqueting està en constant estudi sobre com es mouen els mercats online i informen al client de quines són les estratègies a seguir per a que el seu producte es vengui més respecte dels seus competidors.

Segona planificació

La segona planificació amb les noves tasques està plantejada de la següent manera:

S'han inclòs en la planificació les tasques tant de visualitzador de ventes com de gestió d'estoc. A més a més, al veure que el termini del projecte no es podia assolir pel juny, s'ha aplaçat l'entrega del projecte a setembre. Això ha fet que s'hagi fixat un mes per fer la memòria final del projecte. Podem veure les noves tasques a la figura 5 i el nou diagrama a la figura 6.

El cost per assolir aquestes tasques està calculat per la següent manera:

- Cost per la llicència i utilització de Magento: 0 €
- Cost per la llicència i utilització de Google Docs: 0 €
- Hores de treball aplicades al projecte: s'ha estimat que s'assignaran al voltant de 400 h de treball al projecte. $400 \text{ h} \times 4 \text{ €/h} = \mathbf{1600 \text{ € de cost al projecte per part de l'empresa.}}$



  Task	Effort	Unique ID
▼ 1) Informe prèvi	2w 4d 1h	8
• 1.1) Definició d'objectius	0,5h	2
• 1.2) Definició de les parts	0,5h	9
• 1.3) Redactat del document	1w 3d	10
• 1.4) Revisió i supervisió	1w 1d	11
▼ 2) Anàlisis	6w	73
• 2.1) Anàlisis del problema	2w	74
• 2.2) Anàlisis del importador de Google	2w	75
• 2.3) Anàlisis del comparador	2w	76
• 3) Disseny de classes	1w	79
▼ 4) Desenvolupament	17w 3d	14
▼ 4.1) Google Docs	14w 1d	16
• 4.1.1) Estudi de python i api de google docs	1w	15
• 4.1.2) Connexió i obtenció de documents google docs	1w	17
• 4.1.3) Manipulació del document	2w	18
▼ 4.1.4) Enviament dels productes a la web	5w 1d	71
• 4.1.4.1) Creació dels productes en la e-shop	2w	64
• 4.1.4.2) Actualització dels productes en la e-shop	4d	66
• 4.1.4.3) Actualització dels productes en diferents idiomes	4d	68
• 4.1.4.4) Validació de camps i correcció d'errors	1w 3d	19
• 4.1.5) Sistema de log	2w	21
• 4.1.6) Test	2d	22
▼ 4.1.7) Importació d'imatges	2w 3d	24
• 4.1.7.1) Estudi de com importar les imatges. <i>A pactar amb l'empresa el disseny</i>	1d	25
• 4.1.7.2) Implementació	2w	27
• 4.1.7.3) Test	2d	26
▼ 4.2) Visualitzador de ventes Google Docs	3w 2d	28
• 4.2.1) Estudi d'obtenció i manipulació de les ventes	3d	29
• 4.2.2) Enviament de les ventes a Google	2d	30
• 4.2.3) Manipulació de les ventes	2w	31
• 4.2.4) Test	2d	33
▼ 5) Gestió d'stock	4w 1d	82
• 5.1) Estudi de la gestió d'stock amb Google docs	1w	83
• 5.2) Enviament de la informació a Google	3d	84
• 5.3) Actualització d'stock	2w	85
• 5.4) test	3d	86
• 6) Realització de la memòria	27w 2d 3h	43

Figura 5: Tasques planificació 2

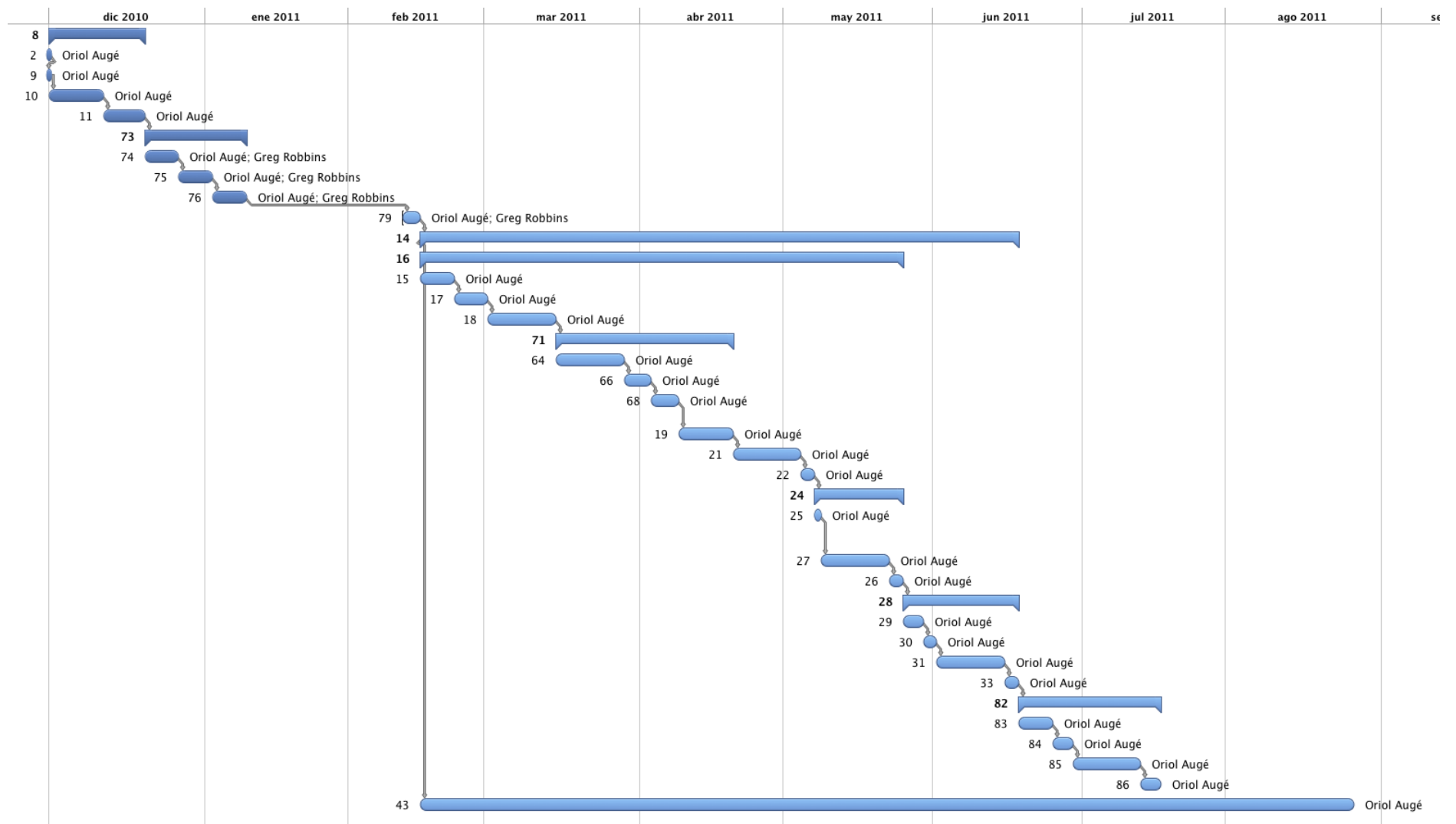


Figura 6: Diagrama de grantt plaificació 2. Cada tasca està identificada amb la seva Id

Capítol 2: Anàlisi i Disseny

En aquest capítol tractarem tots els temes que s'han analitzat i s'han tingut en compte a l'hora de realitzar les anàlisis del disseny del projecte.

Per el que fa a l'apartat d'anàlisi, s'estudiaran els tipus de perfils d'usuaris que pot disposar el nostre projecte, on es resumiran les funcions dels perfils d'administrador i client. També s'analitzaran quins seran els requisits que ha de complir el nostre programa per a que sigui viable i funcional. Els requisits tant poden ser funcionals com no funcionals, deixant aquests requisits com a "guies" per poder dissenyar correctament el projecte.

En l'apartat del disseny del projecte, s'analitzaran els diferents patrons que s'han estudiat per poder crear el nostre projecte. Els patrons escollits han estat MVC, Singleton i Front controller. S'analitzaran aquests patrons, el perquè s'han escollit i la seva unió amb aquest projecte. També s'analitzarà el perquè d'algunes decisions del projecte com ara perquè no hem dissenyat una base de dades per el projecte, els llenguatges de programació utilitzats, entorn de desenvolupament, entre d'altres.

Anàlisi

Perfils d'usuari

El programa ha de ser utilitzat per dos perfils d'usuari:

- **Administrador:** Serà l'encarregat de gestionar i administrar l'aplicació. Les seves taques habituals seran preparar l'entorn per a cada botiga i revisar les importacions.
- **Client (Usuari):** Serà l'encarregat d'introduir els seus productes al Google Docs.

Requisits no funcionals

El programa ha de tenir els diferents requisits:

1. **Modular:** El programa ha de ser fàcil de mantenir, per tant, si hi ha alguna part del codi que ha de ser reescrita, ha de ser fàcil de poder aïllar i corregir la part afectada.
2. **Fàcil adaptació a diferents entorns:** El programa s'ha de poder adaptar a diferents botigues que tindran productes amb atributs i característiques completament diferents.
3. **Programa User-friendly:** El programa ha de ser fàcil de manipular i de treballar-hi tant per al client com per a l'administrador. Per part del client, tota la interfície gràfica serà la de Google. Per tant, es dóna per assolit aquest requisit però, per part de l'administrador, ha de tenir una aparença amigable de preparar l'entorn i treballar amb ell.
4. **Tipus de productes:** El programa ha de ser capaç d'importar els productes simples i els productes configurables de Magento. El motiu per triar aquests dos tipus és perquè són els més utilitzats per part dels clients. Un producte simple és un producte normal sense cap característica especial ni cap personificació per part de l'usuari i un producte

configurable és un producte que engloba productes simples i l'usuari escull quin és el que vol comprar. Un exemple molt clar és una samarreta (configurable) on l'usuari escull la talla que vol (simple). S'especificarà més sobre els tipus de productes en el capítol d'implementació.

Requisits funcionals

El programa ha de complir els següents requisits:

1. **Estabilitat:** El programa, per davant de l'optimització i de la rapidesa en execució, ha de ser estable. S'ha de poder evitar el màxim possible d'errors en la importació de productes.
2. **Detecció d'errors en els productes:** Tot error que no pugui solucionar el programa, haurà de ser fàcilment localitzat per part de l'administrador. El programa ha de tenir algun sistema de notificació d'errors per a l'administrador que realitza les importacions i/o per part del client. A més, si en algun moment falla la importació, s'ha de guardar un registre o un sistema de log per saber on ha fallat i poder localitzar més fàcilment el perquè.

Si es detecta un error en un producte que ha introduït el client degut a un paràmetre incorrecte o algun altre error, el programa ha de continuar important tots els altres productes que siguin correctes en la mateixa execució del programa o la següent, indicant al client que el producte té un error. Aquest plantejament està pensat per poder gestionar el nou model de negoci que ha plantejat l'empresa sobre la gestió de botigues online amb Google Docs.

3. **Gestió d'estoc:** La gestió d'estoc és una acció que només ha de ser actualitzada en dos ocasions:
 - a. Crear el producte.
 - b. Quan el client demani explícitament al programa que actualitzi l'estoc.

Aquests motius estan justificats ja que la gestió d'estoc és una variable que gestiona Magento i no es pot garantir que el valor que consta a Magento sigui el mateix que es va introduir a Google quan es va crear el producte. Només en cas que hagi entrat nou material al magatzem i el client vulgui actualitzar aquest atribut es realitzarà una actualització. En cas contrari no s'actualitzarà l'estoc encara que existeixi aquest atribut a Google.

4. **Informe d'errors per a l'administrador:** L'administrador del programa ha de disposar d'un informe d'errors on es vagi especificant quins productes s'han importat correctament, quins s'han actualitzat i quins han donat error i perquè.
5. **Ràpida instal·lació i fàcil manteniment:** El programa ha de ser capaç de realitzar tota la feina possible i realitzar-la de la manera més automatitzada possible. L'administrador ha de dedicar el mínim temps possible a instal·lar el programa i el programa ha de ser capaç de detectar els errors que faci el client a l'introduir dades.

Per tal que la feina d'instal·lació sigui molt ràpida. S'ha pensat que el programa ha de poder crear el document Google Docs si no està creat prèviament. Tanmateix, ha de ser capaç de configurar el document automàticament. L'acció de configuració del document és la creació de la capçalera del document, on s'indica al client quins atributs ha d'introduir al document i a quina columna.

La capçalera és un camp que sempre haurà de determinar l'administrador de l'aplicació. Això és degut a que el programa s'ha de poder adaptar a tots els entorns possibles i, tot i tenir una capçalera base, sempre podrà tenir variants depenent de les especificacions de la botiga.

A més a més, cada columna estarà lligada a un atribut del producte. Això fa que ha de ser l'administrador qui estableixi la relació columna – atribut. Per agilitzar aquesta feina, l'administrador crearà un csv on, a la primera columna, especificarà el nom de l'atribut a Magento. La segona columna serà el nom de la primera fila de la columna (és a dir, la capçalera). Un cop creat aquest csv, el administrador executarà el programa i es crearà el document automàticament a Google i s'introduirà la capçalera automàticament.

Per acabar, el programa no ha de donar problemes en inserir una nova columna o un nou atribut al document un cop ja ha estat creat. Ha de tenir flexibilitat als possibles canvis que pot demanar el client.

Disseny

Introducció

Abans de desenvolupar una aplicació, és molt important dissenyar correctament el software abans de programar-lo. Si es realitza un bon disseny des del inici, es poden evitar força errors i sempre tindrem una estructura clara del programa.

El disseny també ajuda a la comprensió del projecte a persones alienes. En cas que alguna persona entri a desenvolupar alguna extensió pel projecte, aquesta persona sempre podrà veure el seu disseny i podrà veure amb més facilitats què i quin significat té cada classe. Al ser un projecte en empresa, aquesta part és molt important ja que el projecte ha de poder ser gestionat per altres persones en un futur i s'ha de facilitar al màxim l'adaptació i la comprensió del mateix projecte.

L'estructura del software es basarà en un sistema de mòduls independents seguint l'exemple de l'estratègia "divide & conquer". L'objectiu serà aconseguir un nivell d'abstracció lògic i fàcil de comprensió per als desenvolupadors. Aquesta mesura també afectarà en la dependència de les classes del software, ja que es voldrà aconseguir un software que estarà compost d'una sèrie de components de manera que cadascun proporcioni un bon nivell d'abstracció i que una modificació en un d'aquests components tingui un impacte mínim sobre la resta.

Selecció de l'estructura de classes

Per a la realització d'aquest projecte s'ha escollit el patró **Model Vista Controlador** (MVC) [MVC] per a la distribució de les diferents classes en els seus respectius directoris segons la seva funció. Aquest patró és una arquitectura de software, l'objectiu del qual és separar les dades d'una aplicació, la interfície d'usuari i la lògica del programa en tres components separats (figura 6). Aquest patró és molt utilitzat en aplicacions web.

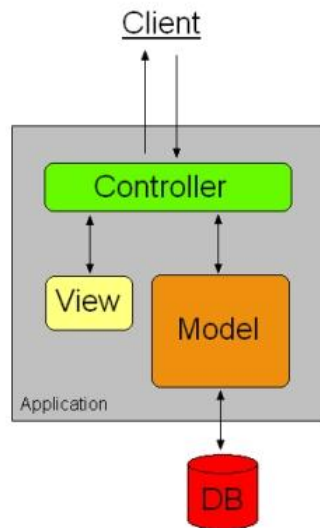


Figura 6: Representació del MVC

Descripció del patró

- Model: Representa tota la informació amb la que el sistema opera.
- Vista: Representa la forma en què l'usuari interactua amb el programa.
- Controlador: Representa totes les accions que realitza l'usuari o altres programes. Crea les peticions al model i, probablement, a la vista.

Aquest programa tindrà dos tipus d'interfície gràfica: la primera serà la utilitzada per l'usuari i serà tota la interfície gràfica dels fulls de càlcul de Google Docs. La segona interfície serà la utilitzada per el administrador que serà a nivell de consola. Aquestes interfícies gràfiques no requereix de cap classe ni cap característica que necessiti una classe tipus "Vista". A més, la única instrucció que tindrà actualment l'administrador en aquest programa serà la seva execució i no requereix d'interfície per executar el programa. Per aquest motiu tota la part de Vista del controlador no s'implementarà-

El controlador haurà de ser l'encarregat de definir el procés d'execució i dir als models que obtinguin les dades de Google, les manipulin i importin/actualitzin els productes corresponents. Les accions que rebrà el controlador tant poden ser a través de consola

(execució del programa) com a través de Google (s'ha de deixar oberta la possibilitat que l'usuari pugui decidir quan fer la importació des de Google).

Un altre motiu per utilitzar aquest patró de disseny és que Magento utilitza aquesta mateixa estructura de documents. Per tant, serà molt més fàcil integrar el nostre projecte a Magento en un futur si s'utilitza la mateixa estructura.

El projecte també ha introduït un nou apartat que no està contemplat als model MVC i és la utilització de “*helpers*”. Els *helpers* són classes que no contenen la lògica del sistema com els Models però són necessaris per a la manipulació d'informació que serà utilitzada pel programa. Per exemple, si és necessari una classe que llegeixi un fitxer i retorni el seu contingut, no es veu necessari introduir-ho a les classes corresponents al Model però es necessita per al funcionament del programa. Per tant, s'exclouen dels models i s'introdueixen en els *helpers*. A més, aquestes classes són genèriques i es poden utilitzar per altres projectes. Un altre exemple també pot ser una classe genèrica de manipulació d'una base de dades que creï i gestioni totes les *queries*.

Els *helpers*, al ser classes genèriques, s'han enfocat seguint el patró ***singleton*** [singleton]. Aquest patró té la característica de restringir la creació d'una classe més d'un cop, és a dir, que la classe *singleton* només tingui una sola instància i proporcionem un sol punt d'accés global en aquesta.

El patró *singleton* s'implementa creant en la nostra classe una instància de l'objecte només si no n'existeix cap. Això es pot regular creant el constructor de la classe com a privat. El motiu més habitual d'utilització d'aquest patró és que les classes controlen l'accés a un recurs físic únic (com, per exemple, el ratolí, un fitxer, una base de dades, etc). També és útil per quan una classe ha de ser accessible per a tots els diferents objectes de l'aplicació (com en el nostre cas).

Un altre apartat que s'ha tingut en compte a l'hora de dissenyar el sistema és que el programa s'ha de poder aplicar a molts tipus de botigues amb característiques diferents. Per poder guardar la configuració de cada botiga, s'ha plantejat la creació de diferents fitxers tipus .ini. Aquests fitxers estaran al directori *sites* i cada fitxer .ini representarà la configuració d'una botiga. Això s'ha plantejat així degut a que cada execució del programa agafarà la informació del fitxer ini corresponent al que es vol executar (variables d'accés, identificador del google docs on hi ha la informació, ruta del fitxer de log, etc). Per tant, el *controller* de l'aplicació ha de ser l'encarregat de gestionar aquesta informació.

Patró “Front Controller”

En aquest punt de la memòria s'ha comentat que tindrem classes categoritzades com a Models i *helpers*, a més, tindrem el *controller* que serà l'encarregat de definir les accions que farà el programa un cop s'executi i els fitxers ini on s'estableixen els paràmetres de configuració de cada botiga. Amb això es planteja el problema de com indiquem al controlador quin fitxer ini ha de carregar sense haver-ne de modificar el codi.

Per solucionar aquest problema s'ha decidit utilitzar el patró **front controller** [front-controller]. Aquest patró és molt utilitzat en el món del desenvolupament web i la seva funcionalitat és que totes les peticions arribin sempre al mateix lloc (és a dir, a un controlador genèric). Aquest controlador és l'encarregat de distribuir les diferents accions al controlador més indicat per realitzar l'acció. Si apliquem aquest patró al nostre programa, ens obligarà a tenir un controlador mestre que serà el que s'executarà quan es vulgui iniciar el programa. Quan s'inicia el programa, aquest controlador mestre se li especificarà quin controlador ha d'executar i quin fitxer .ini ha de fer servir el controlador. L'estructura d'execució del programa serà la següent:

- Index.php <controlador_especific> <.ini_del_site>

El fitxer index.php serà el nostre controlador “mestre” i els paràmetres que rebrà seran el controlador i la ruta del fitxer de configuració. Això també permetrà poder cridar el nostre controlador amb diferents fitxers de configuració o tenir la possibilitat de poder crear més controladors per fer funcions més específiques que es requereixin en un futur.

A més, com es pot veure, totes les peticions hauran de passar sempre pel fitxer index.php, complint perfectament el patró *front controller*.

Diagrama de l'estructura del programa

Un cop definits els patrons que utilitzarà el programa, exposarem un diagrama amb l'estructura del programa amb les funcions bàsiques i la distribució que tindrà a priori (figura 7).

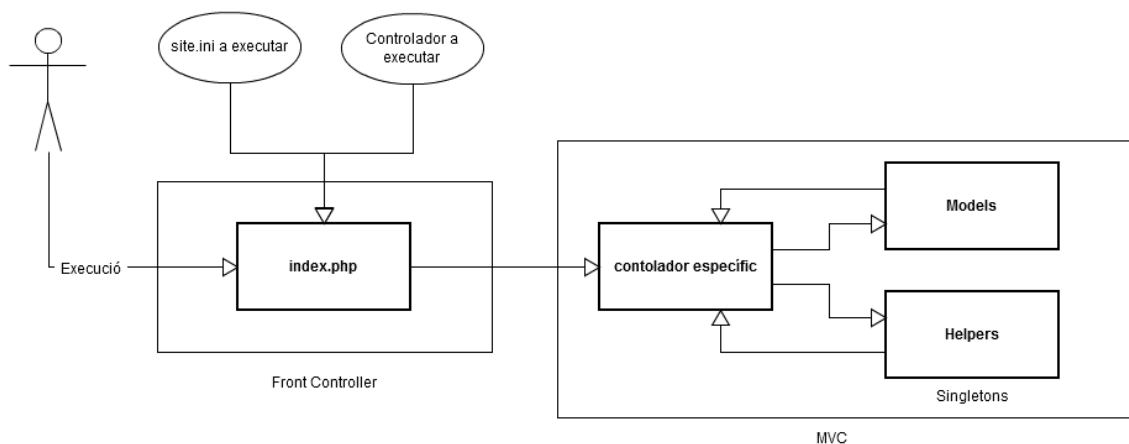


Figura 7: Diagrama de flux de l'aplicació

Com es pot veure en el diagrama, s'ha especificat on s'utilitzen els diferents patrons de disseny.

Selecció del llenguatge de programació

Aquest programa ha sofert canvis en el llenguatge de programació. Primer es va pensar que la realització del programa es podia fer amb Python però, posteriorment, es va haver de reescriure per tenir tot el programa en PHP. Tot i que ja es van explicar en el Capítol 1 els

motius del perquè es va començar en Python i que en el capítol d'implementació es van explicar els motius d'haver hagut de reescriure el codi, a continuació farem una explicació dels dos llenguatges, els seus avantatges i l'adaptació a l'entorn on ens trobem.

Llenguatge de programació PHP

PHP [php] és un llenguatge sotmès a una llicència *open source* utilitzat per a la generació de pàgines web dinàmiques. Les principals característiques de PHP són la seva facilitat d'ús, la quantitat d'eines disponibles per a complementar-lo, la gran comunitat d'usuaris que programen en aquest llenguatge convertint-lo en un dels més utilitzats al web i finalment l'absència de llicències privatives. Al reescriure el codi, es va fer servir la versió 5 del llenguatge, actualment la més utilitzada. Finalment cal dir que PHP s'executa en el servidor remot sota Apache. Apache és un servidor web HTTP de codi obert i gratuït líder mundial en el camp dels servidors i amb una quota de mercat de quasi el 70%. Per tant, la nostra aplicació serà portable fàcilment a la gran majoria d'aplicacions sense haver de modificar-la. Per poder entendre d'una manera més visual la interacció entre apache i PHP, podem analitzar la fiura 8.

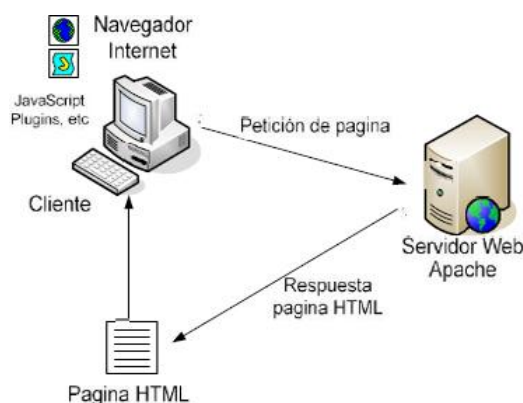


Figura 8: Esquema comunicació Apache + PHP

Llenguatge de programació Python

Python és un llenguatge creat a principis dels anys 90 per Guido van Rossum. Es caracteritza per ser un llenguatge interpretat, és a dir, necessita un programa entremig anomenat intèrpret per poder executar el codi, és multi plataforma i es caracteritza per la seva gran flexibilitat davant d'altres llenguatges com el C, C++, etc. Tot i ser un llenguatge de tipat dinàmic, no es permet tractar una variable de tipus *string* com si fos un número. Altres llenguatges de programació s'adaptarien al nou tipus però això pot crear errors.

La gran diferència entre Python i altres llenguatges és la seva sintaxi, molt semblant al llenguatge natural i la gran quantitat de llibreries de que disposa. A més, tot i no ser un llenguatge compilat, els seus codis generen un fitxer .pyc o .pyo (*bytecode* optimitzat) sent aquests codis els que agafa el intèrpret i fent que la seva execució sigui molt més ràpida que tornar a interpretar el codi en cada execució. A més, és un llenguatge orientat a objectes. Si es

vol aprofundir en el llenguatge de programació Python, es recomana el enllaç [python] de la bibliografia.

L'únic inconvenient que veiem a Python és que hi ha molts conflictes entre les diferents versions de les llibreries o del mateix Python. Això pot fer que un programa que funcionava perfectament deixi de funcionar per una actualització i que sigui molt difícil de solucionar degut a la barreja de paquets que té el sistema de l'usuari. Per solucionar aquest problema, es pot fer servir el programa **virtualenv** [virtualenv]. Aquest programa crea un entorn virtual instal·lant els paquets de Python bàsics i aïllant tots els altres que s'instal·lin en el sistema de l'usuari. Aquests sistema s'ha utilitzat en un entorn Linux i permet portar molt fàcilment una aplicació i els seus paquets a altres sistemes sense haver d'instal·lar totes les dependències necessàries per l'execució.

Necessitat d'una base de dades

Al fer l'anàlisi del projecte, es va pensar en crear una base de dades amb informació dels productes que hi ha a Google docs. El per què d'aquesta opció era intentar reduir el nombre de crides a Google i només demanar la data d'actualització del fitxer. En cas de ser la mateixa que l'última vegada que es va obtenir tota la informació, no es demanen tots els productes i es prenen els atributs de la base de dades com a vàlids. Això es va veure innecessari degut a que l'obtenció del *feed* per part de Google s'ha aconseguit que sigui relativament ràpid. Inicialment es va intentar obtenir tota la informació dels productes cel·la a cel·la depenent de la informació que requeria el programa però això resultava molt costós de temps. Després, es va passar a demanar la informació completa en una sola crida i va augmentar molt el rendiment. Depenent de la quantitat de productes que estiguin a Google, el temps d'espera pot ser major o menor. Per exemple, en cas de 2000 productes es pot tenir un temps d'espera entre 5 o 8 segons.

També es va veure innecessari la base de dades ja que tota la informació es veuria repetida 3 vegades. Tindríem Google com a font d'informació, la mateixa informació a la base de dades intermèdia i, per acabar, la mateixa base de dades de Magento.

Consultes a la base de dades de Magento

Per a la realització de les consultes a la base de dades de Magento, s'ha decidit fer servir la llibreria SOAP [soap] que disposa Magento. Aquesta decisió ve justificada per:

- El programador ja està familiaritzat amb aquesta tecnologia per projectes anteriors. Això agilitzarà la programació del projecte i es podrà tenir un prototip molt més ràpid.
- S'ha plantejat amb l'empresa que seria molt útil que aquest projecte es pugui executar des d'un ordinador però manipulant la base de dades d'un altre ordinador. Això pot permetre fer tests en entorns locals, en entorns local <--> servidor, servidor <--> servidor, etc. Si no s'utilitzés la tecnologia SOAP, les execucions s'haurien de fer sempre en el mateix servidor.
- Totes les funcionalitats ja implementades amb SOAP no s'han de programar, fent que estalviem temps de programació.

El desavantatge d'utilitzar SOAP en lloc de les funcions de Magento és que el rendiment es veurà afectat i es veurà reduït. Es va fer una reunió amb l'empresa i es va fixar que la prioritat actual era tenir un prototip funcional i estable del projecte ja que, com a mínim, es volia eliminar l'actual sistema d'importació i substituir-lo per aquest al més aviat possible. Entenem que la tecnologia SOAP és una tecnologia antiga i no és tan òptima com utilitzar funcions o classes pròpies de Magento però, si s'ha de millorar el codi en un futur, l'empresa prefereix deixar-ho com a un projecte de futur o com una segona versió.

Perquè SOAP i no XML-RPC

Magento té dos tipus de connexió *web-service*. Les consultes es poden fer tant amb SOAP com amb XML-RPC [xml-rpc]. El motiu per decantar-se per SOAP i no utilitzar XML-RPC ha estat simplement l'estructura de SOAP. SOAP té una estructura definida en el WSDL (*web services description language*) que fa que d'interacció amb el *web-service* es faci amb un "llenguatge neutre". Es pot programar amb qualsevol llenguatge de programació que suporti SOAP i, en crear la connexió, cal basar-se en les funcions i mètodes que estan definits al WSDL. És allà on es defineixen les funcions, tipus d'entrada i de sortides de les funcions, etc. (veure figura 9)

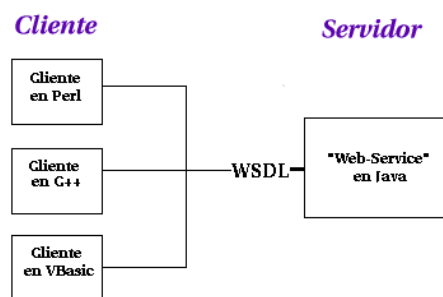


Figura 9: Escenari SOAP

El XML-RPC no té aquests "diccionaris" i cal conèixer abans de programar quines són les funcions que estan disponibles per al *web-service* i com interactuar amb elles. A més, com que el plantejament inicial és que el programa es desenvolupi amb 2 llenguatges diferents, es veu més òptim fer la crida de funcions via SOAP.

Entorn de desenvolupament

Per programar aquest programa s'ha utilitzat l'entorn VIM (figura 10) [vim]. Vim és una versió millorada de l'editor de text vi, present en tots els sistemes UNIX. Vim és un editor *open source* que es diferencia de la resta d'editors per l'opció de poder alternar en realitzar diferents operacions, es a dir, té un mode per introduir instruccions i un altre per introduir text. És un editor que permet un control absolut sense necessitat d'utilitzar el ratolí, fent una programació més còmoda per al programador i s'executa a nivell de consola.

Una de les raons d'utilitzar aquests sistema és que els servidors de l'empresa no disposen d'interfície gràfica. Per això, cal un entorn de desenvolupament que s'executi a nivell de consola i permeti fer canvis al servidor quan sigui necessari. A més, vim permet la instal·lació de diferent mòduls que li aporten moltes funcionalitats extres.

```

72 }
73
74 public function getAttributeId()
75 {
76     return $this->attributeId;
77 }
78
79 /**
80  * Funció per eliminar espais i fer més fàcil les comparacions entre el contingut
81  */
82 public function clearSpaces($data)
83 {
84     return str_replace(' ', '', $data);
85 }
86
87 /**
88  * Funció per eliminar espais i tornar les lletres en minúscules per evitar possibles errors del client i fer
89  * més fàcil les comparacions
90  */
91 public function clearSpacesToLower($data)
92 {
93     return str_replace(' ', '', strtolower($data));
94 }
95
96 /**
97  * Creació d'un array on no hi hauran espais, útil per els productes associants, categories, atributs
98  * configurables, etc.
99  */
100 public function explodeClearSpaces($data)
101 {
102     return explode(' ', str_replace(' ', '', $data));
103 }
104
105 private function setAttributeOptions($attribute)
106 {
107     $result = null;
108     foreach ($this->attributeList as $valuesAttribute)
109     {
110         if ($valuesAttribute['code'] == $attribute)

```

Figura 10: Entorn de desenvolupament VIM

Git: Control de versions pel projecte

Per poder gestionar els diferents fitxers d'un projecte correctament és recomanable utilitzar un programa que pugui gestionar les diferents versions del projecte. Amb aquests programes, podem detectar més fàcilment quins són els canvis que s'han realitzat en el projecte en un moment donat, es pot desfer un canvi, recuperar una versió anterior, permet fer algun test i, un cop s'ha desenvolupat, deixar el projecte de la mateixa manera que estava abans ja que el canvi no s'aplicarà, etc.

Per poder gestionar les diferents versions del nostre projecte s'ha escollit el programa Git . Git és un programa gratuït de control de versions molt utilitzat en entorns Linux i més lleuger i ràpid que el programa Subversion. La seva instal·lació és molt senzilla: només cal instal·lar el programa i utilitzar el comandament "git init" en el directori arrel del programa per començar a treballar git. Tot i que normalment aquests programes dipositen el codi en repositoris perquè varis membres d'un projecte puguin descarregar-se el codi i treballar en ell, git també permet el control d'una manera local i sense repositori extern, emmagatzemant tots els canvis a la carpeta .git que es crea al instal·lar el programa en el nostre projecte. Aquesta eina serà la utilitzada per emmagatzemar tots els canvis que s'aniran desenvolupant.

Tot i que en aquesta memòria no s'explicarà el funcionament de git, es recomana llegir l'enllaç [git] de la bibliògrafa per saber els les funcions bàsiques de git (per exemple, git add, git commit i git push) o per ampliar la informació.

Diagrames de seqüència

A la figura 11, mostrarem un diagrama de la seqüència o el procés que ha de realitzar el programa per poder importar/actualitzar els productes a la botiga *online*.

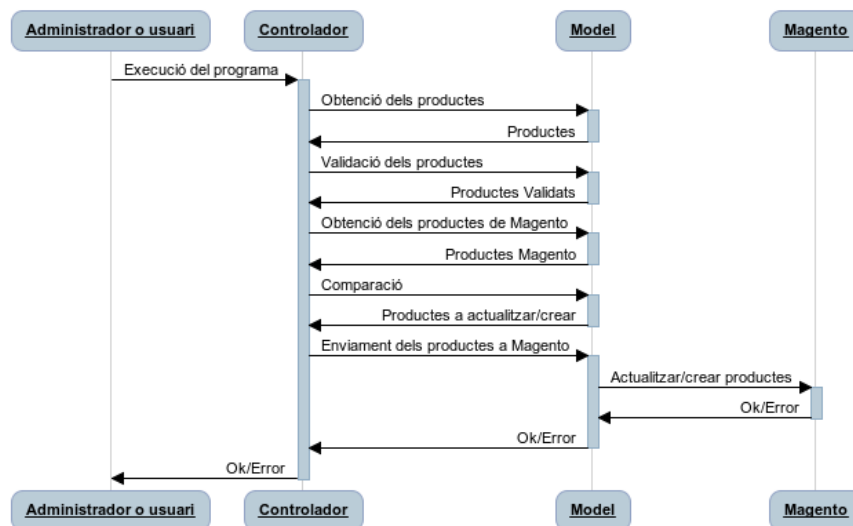


Figura 11: Diagrama principal de seqüència de l'aplicació

En el diagrama de la figura 11 podem veure les principals accions que s'han analitzat en el programa. Com que aquest programa està enfocat en que l'administrador no hagi de fer moltes accions per executar el programa, la única instrucció que rebrà el controlador del programa serà el inici de l'execució del programa. Un cop s'ha executat el programa, aquest obtindrà els productes que s'han introduït al full de càlcul de Google Docs. Un cop disposem dels productes, el programa haurà de validar el contingut per saber que els productes són aptes per la importació. Al saber amb seguretat que els productes són aptes, el programa obtindrà els productes de Magento per saber si s'han d'importar (producte nou) o s'ha d'actualitzar (producte ja existen a Magento). Un cop disposem de les dos llistes de productes, es comparen els productes de Magento i Google per saber si han modificat o no en cas que s'hagin d'actualitzar. Per finalitzar, el programa importa o actualitza els productes i finalitza la seva execució.

Capítol 3: Implementació

En aquest capítol es tractarà el que fa referència a la implementació i al resultat final del projecte. Es farà una explicació de tots els problemes que s'han trobat a l'hora de desenvolupar el programa i tots els punts que han afectat a la planificació del projecte, endarrerint o modificant les tasques inicials.

Els temes més importants que es tractaran seran el perquè del canvi de llenguatge de programació que s'havia establert en la planificació, la justificació de la creació d'un nou mòdul Magento (juntament amb tots les temes que s'han tingut en compte per a la seva creació) i l'estructura final del projecte amb una explicació de les classes/fitxers utilitzats en ell. També s'analitzarà els canvis a la lògica del sistema, el fet d'haver d'implementar l'actualització de productes en diferents idiomes i el llistat de comandes que s'ha creat.

Google Docs

Per desenvolupar tota la comunicació entre el nostre programa i Google, s'ha utilitzat l'API de Google Docs [api-google] i l'API de Spreadsheet document [api-google2]. Aquestes dues llibreries que conté el Zend Framework són les encarregades de la manipulació de documents (Google Docs) i la manipulació de fulls de càlcul (Spreadsheet).

Passar de Python a PHP

Un dels problemes que s'ha trobat a l'hora d'implementar aquest projecte és la comunicació entre Magento i Python (figura 12). L'idea inicial que es va pensar va ser que Python s'encarregava solament del que és la descarrega i manipulació de dades Google Docs i, per altra banda, tota la manipulació de dades de Magento es realitzava amb PHP (ja que Magento està programat amb PHP).

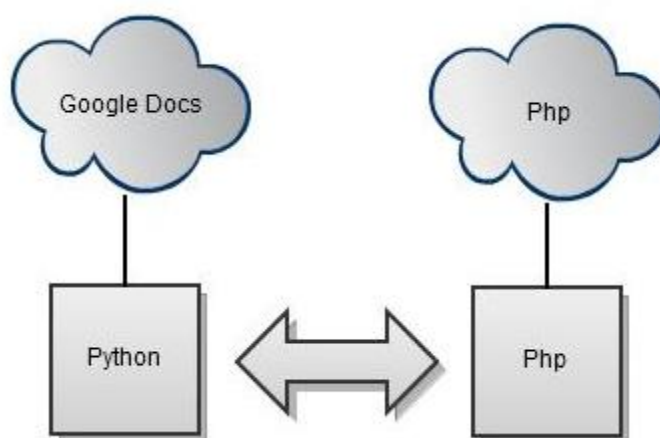


Figura 12: Escenari inicial de l'aplicació on conviuen els dos llenguatges de programació

El programador va crear tota la part referent a Python, amb això, va poder estudiar les diferents funcions de l'API de Google, va poder ampliar els seus coneixements en programar en el llenguatge de programació Python i les seves estructures. Les classes que es van crear van ser:

- SpreadsheetClient

Aquesta classe és l'encarregada de manipular la informació del full de càlcul. Els mètodes que es van implementar van ser:

- Connect: Aquesta funció es connectava a Google docs i buscava, entre tots els documents, el que s'havia especificat amb el nom del document. En cas que no trobés aquest document, la funció en creava un amb aquest nom.
- downloadDocumentsContent: Funció que es descarrega tota la informació que conté el document.

- makeDictionari

Aquesta classe és l'encarregada de llegir un csv on hi ha la relació entre atribut de Magento i atribut (columna) en Google. Les funcions que té aquesta classe són les típiques per obrir i llegir un document csv.

- googleClient

Google té una classe per crear documents "en general" i una altra classe més específica per poder manipular la informació del document. Aquesta és la classe que encapsula les funcions genèriques com ara crear un document. Els mètodes que es van implementar van ser el connect (era necessari una altra connexió amb aquesta classe) i el mètode create.

Un cop creades aquestes classes, es va començar a crear el controlador. Aquí va ser on va sorgir el problema ja que el controlador havia d'interactuar amb dos llenguatges: PHP i Python. El controlador ha de ser l'encarregat d'agafar tota la informació que té Google (Python) i passar-la a Magento (PHP). Per fer les crides SOAP correctament, era necessari tenir el codi en PHP ja que SOAP interactua molt millor amb la versió PHP. Amb Python es van trobar força errors amb les consultes SOAP de Magento que feien necessari la utilització de *parsers* de les respostes del web service i funcions per adaptar la informació a les nostres necessitats, això va fer arribar el punt que totes les crides SOAP amb Python que es realitzaven eren incòmodes de processar i d'interpretar. Tot i això, Magento té una versió 2 de SOAP que intenta evitar aquest error però en altres projectes es va poder comprovar que té varis errors que, tot i que segurament es corregiran en futures versions de Magento, en el moment de desenvolupar el projecte no es disposa de temps ni per corregir nosaltres els errors ni per esperar les actualitzacions de la plataforma Magento. L'empresa va decidir la utilització de PHP per treballar amb SOAP ja que seria més beneficiari i suposaria estalviar temps en la correcció d'errors i programació.

Veient que la interacció amb Magento s'havia de fer amb PHP obligatòriament en aquell moment, es va procedir a crear un controlador (fet amb Python) que cridava a un fitxer PHP

amb la funció `os.system('php -f importProducts.php' + args)`. El problema que es va veure és que el fitxer PHP rebia tota la informació procedent del controlador com a arguments i com a un sol *string*. Es va estudiar i es va veure molt ineficaç ja que el controlador Python rebia totes les dades amb un *array* de dades i, al passar a *string*, es perdia la consistència i tots els avantatges que pot aportar treballar amb *arrays*. Es va valorar l'opció de crear un *parser* del mateix *string* per, després, tornar a convertir aquest *string* en *array* o, també, crear fitxers temporals perquè els dos programes es poguessin comunicar però es va veure més òptim i més fàcil passar el codi ja escrit a PHP i continuar des del mateix punt. Igualment, es pot considerar que amb aquest codi que el programador havia fet ja s'assolia una part dels objectius que era aprendre i saber treballar amb Python tot i que no amb el nivell que es volia adquirir.

Creació d'un mòdul Magento

Un dels requisits que va establir l'empresa a l'hora de crear aquest projecte és que s'importessin els productes tipus simples i els productes tipus configurables. L'empresa és conscient que hi ha més tipus de productes però aquests són els més utilitzats a nivell de client i, amb aquests 2 productes, s'engloba el 90 % de les importacions que es realitzen.

Un producte simple és un producte normal sense cap característica especial ni cap personificació per part de l'usuari i un producte configurable és un producte que engloba productes simples i l'usuari escull quin és el que vol comprar. Un exemple molt clar és una samarreta (configurable) on l'usuari escull la talla que vol (simple).

El programador no va trobar gaires complicacions crítiques mentre implementava l'actualització/creació de productes simples però, en intentar crear/actualitzar un producte configurable, es va veure que Magento no disposava de cap mètode SOAP que enllacés els simples amb el seu configurable. Si no es podia crear la relació entre els diferents productes, la importació es feia "poc útil" i "ineficaç". L'empresa i el programador pensaven que Magento tenia aquestes funcions i, al veure que no estava programada, es va veure convenient crear un mòdul nou que inclogués aquestes funcions ja que, tot i que són necessàries per continuar amb l'importador, també podien ser molt útils per actualitzacions de botigues que ja tenien els productes importats i s'havia de fer algun manteniment o algun canvi massiu en els productes (sense necessitat de fer servir l'importador per actualitzar els productes).

El mòdul que s'ha creat s'ha fet per suplir la falta de funcionalitats per enllaçar els productes configurables amb els seus simples. Per fer-ho, s'ha seguit els següents passos i s'ha utilitzat la filosofia Zend Framework.

Zend Framework

Zend Framework [zend-framework] és un framework de codi obert per desenvolupar aplicacions web i serveis web amb la tecnologia PHP5. L'estructura de tots els elements que componen aquest framework és 100% orientat a objectes i, a més, cada component està construït amb una baixa dependència d'un amb l'altre. Això fa que cada component es pugui fer servir de manera individual.

Zend Framework està enfocat a donar una robusta implementació del model MVC, donant moltes facilitats als desenvolupadors per tenir una gran llibreria de funcions per poder desenvolupar un projecte web amb una facilitat i robustesa molt important.

El principal patrocinador del projecte Zend Framework és Zend Technologies, però moltes empreses han contribuït al desenvolupament d'aquest projecte. Google, Microsoft i Strikelron han aportat importants parts a Zend per poder ser el que Zend Framework és ara, un dels més coneguts i utilitzats frameworks PHP del moment.

Abans de començar a explicar la estructura del Zend Framework per poder entendre Magento, fem un resum de què és el model MVC.

MVC (Model – Vista - Controlador)

Com s'ha explicat en el capítol d'introducció, el MVC [MVC] s'ha convertit en un estàndard en el disseny d'aplicacions web modernes. El seu punt fort és la separació de les classes depenent dels interessos o de les funcionalitats de cada una i, permet agrupar les diferents parts de l'aplicació que es relacionen amb la presentació, el que implementa la lògica del "negoci" i el que accedeix a la base de dades. Molts desenvolupadors troben molt útil aquesta separació per ajudar a mantenir el codi ben organitzat, especialment quant més d'un desenvolupador es troba treballant en la mateixa aplicació (figura 13).

Les classificacions que es fan a nivell de codi són: Model, Vista i Controlador

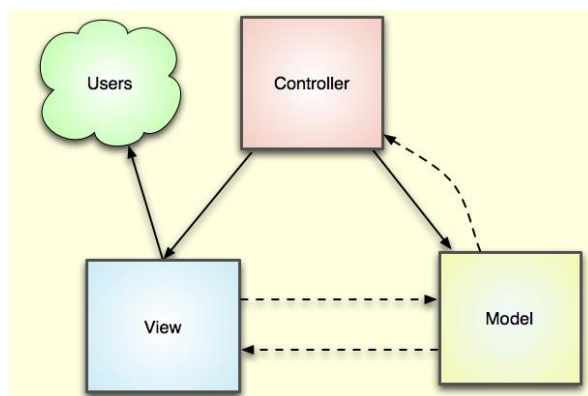


Figura 13: Representació del MVC

Quina és la funcionalitat de cada component?

- **Model:** És l'encarregat de les funcions bàsiques de l'aplicació, incloent les rutines d'accés a dades i a la lògica del "negoci".
- **Vista:** S'encarrega de generar el que es presenta a l'usuari a partir de les dades que rep del controlador, al mateix temps, recull les dades que donen els usuaris. És en les aplicacions web on es trobarà tot el codi html i codi visible.

- **Controladors:** Són els que uneixen el patró. Segons el que demana l'usuari i altres variables decideixen si s'ha d'executar un altre controlador o s'han de manipular les dades del model per, després, assignar-li el resultat a una vista. És recomana tenir el controlador el més net possible per a una millor lectura.

Arquitectura de Magento

Magento està enfocat amb la mentalitat que cada negoci és diferent, és a dir, que cada usuari té necessitats diferents i vol modelar el programa a les seves necessitats. Per aquest motiu, Magento està desenvolupat amb la tecnologia Zend Framework, perquè permet incloure i crear mòduls amb molta facilitat i sense crear dependències entre ells.

Per poder fer una adaptació de les necessitats del negoci en Magento hi ha varies maneres (unes més recomanables que altres depenent de les necessitat). La primera opció i la més ràpida és modificar les funcions de Magento perquè puguin fer les funcionalitats desitjades. Una altra manera, tot i que no es creen directament les noves funcionalitats, és descarregar-se una extensió de Magento creada per a la comunitat. Magento té una gran comunitat molt activa que cada dia està pujant nous mòduls, tant de pagament com gratuïts. És possible que algunes de les funcions que es necessiten ja les hagi implementat algú altre i sempre és recomanable buscar una mica per veure si ja hi ha alguna extensió que fa el que hom pugui necessitar. A més, també es poden adaptar els mòduls a les necessitats de la mateixa manera que es pot fer amb el codi de Magento. El problema és que quant hi ha una actualització, les modificacions es perdran o no es podrà actualitzar.

L'arquitectura del codi de Magento està en el directori `app/code`. Aquest directori té 3 carpetes que estan distribuïdes per *core*, *community* i *local*:

- **Core:** Comprèn el codi original de Magento. És recomanable que MAI s'apliquin modificacions en aquesta part ja que, en qualsevol actualització, es sobreescriurà i es perdrà tot el que s'hagi fet, fins al punt que la web deixaria de funcionar. A més, si es realitzen modificacions en el *core*, és recomanable no actualitzar mai Magento, havent de mantenir una versió antiga.
- **Community:** Comprèn els fitxers de les extensions de la comunitat. Això fa que estiguin aïllades del codi font de Magento i que estiguin fàcilment localitzables.
 - Les extensions són fitxers agrupats que afegeixen funcionalitat a la botiga. La instal·lació es pot fer tant copiant els fitxers en aquesta carpeta manualment o des de el backoffice a través del MagentoConnect. El que Magento anomena extensions en realitat són mòduls.
- **Local:** Comprèn les pròpies extensions, modificacions, etc.

La prioritat de Magento per saber quina extensió agafar és: local, si no està a local mira a community i, si no troba cap extensió, agafarà el comportament del core (seria l'equivalent al comportament "default"). Per més informació recomana el manual [magento] de la bibliografia.

També hi ha diferents regles que s'han de saber per poder crear un mòdul Magento. Aquestes regles són les que estan dictades per el Zend Framework i són per especificar la disposició i situació del mòdul. Per exemple, els noms dels mòduls o classes no poden tenir “barres baixes” ja que, la barra baixa, és el separador de la ruta entre els diferents nivells de directoris. Aquesta regla està en tots els projectes basats amb Zend Framework, ja que és així com Zend interpreta les rutes i va a buscar els fitxers. Per saber més sobre la programació amb Zend Framework es pot consultar l'enllaç [zend-framework2] de la bibliografia.

Estructura de fitxers

A Magento tots els mòduls han de complir una estructura de fitxers per poder ser interpretats. El nostre mòdul s'ha implementat en la carpeta “local” ja que ara mateix és un mòdul d'ús intern que no està alliberat per l'ús de la comunitat i perquè no es vol sobreescriure o que entri en conflicte amb l'apartat “core” de Magento. L'estructura que s'ha de complir es pot veure a la figura 14 que s'ha extret del manual [magento] de la bibliografia:

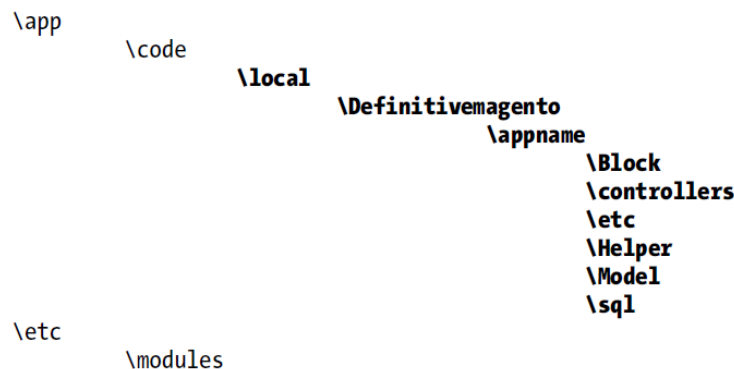


Figura 14: Representació de l'estructura de fitxers d'un mòdul Magento

On:

- \app\code\local\Definitivemagento: Nom del directori arrel del nostre projecte. Normalment és la companyia.
- \app\code\local\Definitivemagento\appname: Nom de l'aplicació o de la utilitat del mòdul. A més, si l'objectiu és crear diferents mòduls sobre la mateixa companyia, poden estar tots dintre del directori d'aquesta i crear diferents mòduls interns.
- \app\code\local\Definitivemagento\appname\Block: Aquest concepte ve del patró MVC. Representa la capa de la vista. El Block conté la informació que es mostrarà en el html, en els CMS blocks o en els phtml.
- \app\code\local\Definitivemagento\appname\Controllers: Aquí s'introduiran les accions del mòdul. S'ocupa de tota la lògica i passa les dades entre els models i els blocks.
- \app\code\local\Definitivemagento\appname\etc: Aquest directori conté els fitxers que necessita Magento per interpretar el mòdul i saber com interaccionar amb ell. Normalment té un fitxer config.xml que serà el que llegirà Magento per saber què conté aquest mòdul i què fer amb ell.

- \app\code\local\Definitivemagento\appname\Helper: Els *Helpers* serveixen, en essència, per crear aplicacions o per tenir rutines que es fan servir en vàries parts del mòdul. Destaquem que en el nostre codi també s'ha creat una part anomenada *helper* que té aquest objectiu.
- \app\code\local\Definitivemagento\appname\Model: Aquí és on es defineix la lògica d'interacció amb la base de dades. És aquí on es defineixen les crides per buscar, guardar o obtenir informació de la base de dades.
- \app\code\local\Definitivemagento\appname\sql: Encara que el nom del directori pot portar a alguna confusió, aquí no és on es guarden les crides sql a la base de dades, en aquest directori definim la creació de les taules de la base de dades personalitzades que pot necessitar el mòdul i, també, es guarden les actualitzacions de les taules que es necessiten per poder actualitzar el mòdul en futures actualitzacions.
- \etc\modules: Aquest és un directori que es troba fora del conjunt del paquet. És el directori on es notifica a Magento quins mòduls o extensions estan instal·lades.

El resultat del nostre mòdul ha estat el següent (figura 15):

```

~Etailers/
  ~Soap/
    ~etc/
      -api.xml*
      -config.xml*
      -wsdl.xml*
    ~Model/
      ~Product/
        ~Api/
          -V2.php*
        ~Attribute/
          ~Api/
            -V2.php*
            -Api.php*
          ~Link/
            ~Api/
              -V2.php*
              -Api.php*
            -Api.php*
      -Etailers_All.xml*

```

Figura 15: Fitxers del mòdul Etailers SOAP

El fitxer Etailers_All.xml és el fitxer que ha de ser instal·lat a la carpeta \etc\modules. Els altres fitxers es descriuran a continuació:

Com el mòdul ha d'implementar noves funcions SOAP, no és necessari cap Block, Controller, Helper i sql. L'únic que ha de fer aquest mòdul és interactuar amb la base de dades i fer les operacions necessàries en aquell moment. A més, el controlador del programa està apartat i

fora de Magento, fent que sigui el controlador del programa qui decidirà quan cridar les funcions creades i és ell qui manipularà les dades que obtindrà d'aquests models.

S'han creat 3 grups de funcions, una per agrupar les funcions referents a crear, actualitzar i obtenir un llistat de productes, una altra per poder gestionar els *labels* dels atributs configurables i, per últim, una altra per gestionar els productes configurables.

Les funcions per crear/actualitzar i obtenir un llistat s'han realitzat per millorar les pròpies de Magento. Aquestes fan les mateix procediment però s'han adaptat perquè sigui més fàcil pel programa interactuar amb Magento i adaptar-se a ell. Per exemple, si a Magento un producte té un atribut desplegable, la funció pròpia de Magento s'ha de passar la id del atribut perquè l'actualitzi. Amb les nostres funcions, s'ha adaptat de manera que ja no cal la id de l'atribut sinó simplement amb el nom (que és el que s'obté des de Google docs). L'objectiu d'aquestes funcionalitats és poder manipular millor la informació que introdueix l'usuari a Google Docs sense que ell hagi de saber quines són les ids dels seus atributs. Una altra funcionalitat que s'ha modificat és el llistat de productes de Magento. Quan es demana un llistat de productes, Magento retorna la informació bàsica del producte i no tota la descripció completa del producte. Aquest fet feia que cada cop que es volia comparar un producte, s'havia de cridar una funció per obtenir tota la informació.

Les funcions referents als *labels* s'han fet per poder crear i actualitzar *labels* dels atributs. Abans d'explicar el perquè d'aquestes funcions, s'explicarà què és un *label* d'un atribut:

Els productes de Magento estan creats per conjunts d'atributs. Cada conjunt d'atributs pot tenir uns atributs o altres adaptant-se a les necessitats de cada conjunt. Els atributs són les variables de cada producte que contenen informació de cada un. Un exemple d'un atribut pot ser el preu, la descripció del producte, una foto, etc.

Cada atribut pot ser de diferents tipus: un camp de text, una àrea de text, una variable booleana, etc. En aquest punt ens centrarem en els que són de tipus desplegable:

Els atributs de tipus desplegable són aquells que el producte només pot tenir un valor i, a més, dintre d'un rang fixat prèviament dins del mateix atribut. Per exemple, suposem que volem que tots els productes tinguin guardada la seva marca. Com que tenim un nombre finit de marques i, en el cas de l'exemple, un producte només pot ser d'una sola marca, crearem l'atribut marca i l'iniciarem com a atribut desplegable. Un cop definit l'atribut, s'han de crear els *labels* dels atribut. Els *labels* són els valors que pot tenir aquest atribut dintre d'un producte. Si tornem a l'exemple, introduiríem els *labels* Adiddas i Nike, així només constarien aquests dos valors pels productes.

Si al cap d'un temps apareix una nova marca, es gestionaria per backoffice: caldria entrar dintre l'atribut marca i introduir la nova.

L'avantatge d'aquests atributs és que, al tenir un nombre finit d'opcions, es poden fer "filtrables" a nivell de web fent que, al navegar per la botiga en les vistes de la categoria, es

puguin filtrar els productes que veiem per opcions dels *labels* dels atributs desplegable com podem veure a la figura 16.

També existeixen atributs muti-opcions. Tenen les mateixes característiques que els desplegable però permeten que un producte tingui més d'una opció.

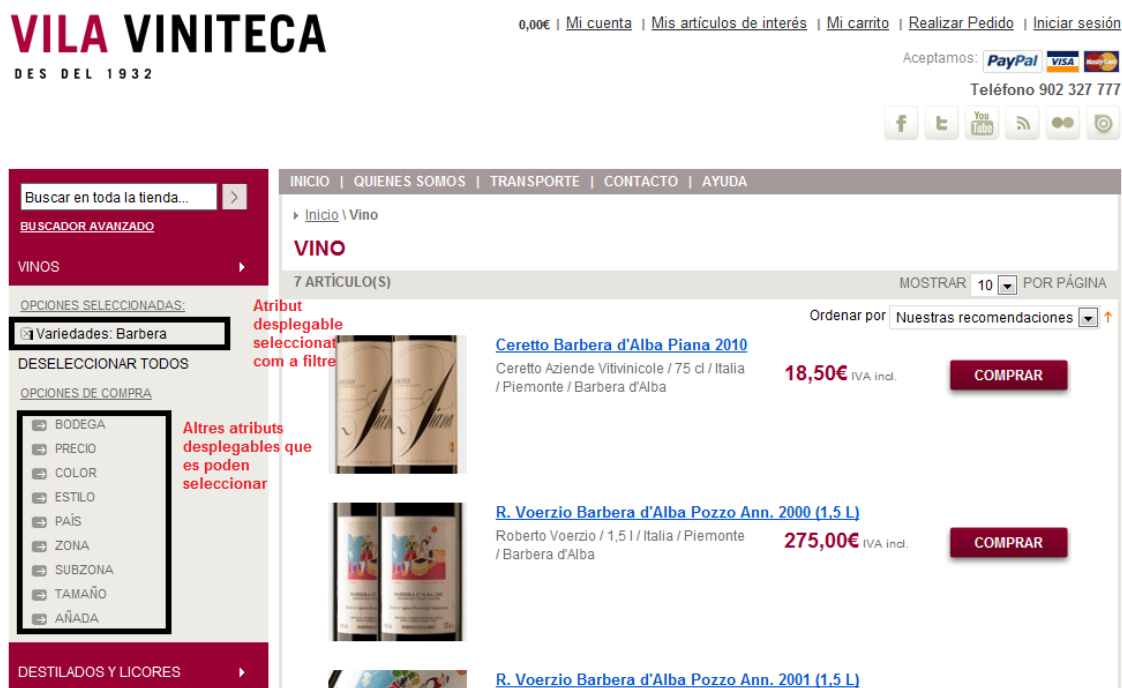


Figura 16: Exemple d'utilització de filtres de la web www.vilaviniteca.com/shop. A l'esquerra, es pot veure emmarcat els filtres.

Les funcions que s'han creat són per poder gestionar els *labels* des dels programes i no des de backoffice. Magento no permet importar nous labels per els atributs si no és per backoffice. Per suplir aquesta mancança de Magento s'ha creat aquesta funció.

Les funcions que estan al directori Link són les que estan encarades a realitzar totes les operacions dels productes configurables.

Els productes configurables són varis productes simples però amb alguna diferència en els seus atributs tipus desplegable. A la vista de la botiga, estan presentats com un sol producte en què pots escollir el valor de l'atribut desplegable que vols comprar, per exemple, si es vol tenir una samarreta en diferents talles, s'han de crear els productes simples cada un amb el valor de la talla associada. Un cop consten tots els simples, s'ha de crear el configurable que agruparà tots els simples i serà aquest l'element que es veurà a la pantalla de la botiga (si així es vol).

El que s'ha creat és una funció que enllaça els productes simples corresponents al configurable indicat. S'enllaçaran sempre i quan aquest configurable tingui l'atribut corresponent a que es poden enllaçar, es a dir, no es podrà enllaçar un producte configurable que no té cap atribut desplegable o un atribut que no tenen relació, com ara el color i la talla.

A més, Magento té l'opció d'incrementar el preu de les opcions de l'atribut en funció del preu base del configurable. Si una samarreta val 10€ i es presenta en les talles L,M,XL, el preu base del configurable és 10, però si tenim l'opció de XXL i, aquesta té un increment de preu de 5€, Magento dóna l'opció d'incrementar el preu del configurable al seleccionar aquesta talla.

El problema d'això és que no és un procés automàtic. Magento no calcula el preu del configurable a través dels simples associats i tampoc calcula l'increment de les seves opcions. El que s'ha fet en aquesta funció és, en relació de la diferència entre el simple i el configurable, configurar automàticament l'increment o el decrement del preu.

Estructura final del projecte

En l'actualitat, el projecte té l'estructura de directoris i els fitxers que es poden veure a la figura 17:

```
/home/oriol/Escritorio/Google D
|~controller/
| | -GoogleImport.php
| | -GoogleImportCategoriesId.ph
| | -GoogleImportImages.php
| | -GoogleImportUpSell.php
| | -GoogleStockList.php
|~helper/
| | -compareArraysGoogleImport.p
| | -log.php
| | -makeDictionaryGoogleImport.
| | -validateGoogleImport.php
|~lib/
| | +Zend/
|~model/
| | -dataObj.php
| | -googleClient.php
| | -importContent.php
| | -MClientGDocs.py*
| | -MClientMagento.php
| | -MClientSoap.php*
| | -spreadSheetClient.php
+sites/
| | +accesoriosdecoches/
| | +actigreen/
| | +bricmania/
| | +canadahouse/
| | +cottet/
| | +dali/
| | +deacampada/
| | +farmacia/
| | +gafasdesol/
| | +garavilla/
| | +tumueble/
| | +vileda/
| | +vivabebe/
| | +vv/
| | +vvoutlet/
| -index.php
| -README
```

Figura 17: Estructura final del projecte i els seus fitxers. També es poden veure alguns directoris on estan els fitxers .ini

Controller: Són els controladors del projecte. S'han dissenyat varis controladors per poder fer diferents operacions que s'han necessitat en alguns projectes o per poder fer alguna operació més ràpida. Els controladors que s'ha creat són:

- **GoogleImport.php:**
Controlador que s'encarrega de crear o actualitzar els productes segons si estan en la web o no i segons els paràmetres que estan guardats a Google Docs.
- **GoogleImportCategoriesId:**
Aquest controlador es va crear per poder fer una relació molt ràpida entre sku <-> id de la categoria on es vol assignar. En la primera versió del controlador GoogleImport.php, l'assignació de categories es feia a través del nom de la categoria. L'usuari introduïa el nom de la categoria on volia assignar el producte en Google docs i, el programa, buscava aquesta categoria i assignava el producte en aquesta. En aplicar el projecte en webs reals es va donar el cas que el nom de la categoria podia estar duplicat (exemple: Pijames -> Home, Vestits -> Home). Quan es va detectar aquest problema, es va crear aquest controlador per poder assignar els productes a les seves categories en relació a la id de la categoria. Actualment, GoogleImport.php accepta tant noms de categories com ids fent que aquest controlador només s'utilitzi per fer assignacions ràpides sense haver de comparar si el contingut del producte ha estat modificat, el producte s'ha creat, etc.
- **GoogleImportImages.php:**
Aquest controlador és l'encarregat d'agafar les imatges dels productes d'un directori i importar-les al seu producte. És aquest el controlador que s'ha creat per poder suplir el problema de com importar imatges. Per funcionar, es parteix de Google Docs, al que s'adjunta una columna anomenada "image" i, allà, s'especifica el nom de la imatge que es vol per al producte (nom del fitxer). Quant s'executa el controlador, agafarà el nom corresponent de la imatge i buscarà aquesta imatge en un directori prèviament especificat. Un cop ha localitzat el fitxer, el programa el relacionarà amb la imatge del producte.

Normalment les imatges es dipositen en un directori FTP que se li facilita al client. A més a més, s'han fixat diferents paràmetres perquè el client pugui escollir quina imatge és la més representativa del producte i quines són les imatges secundàries que aniran a la galeria de fotos del producte:

Els noms de les imatges estaran formats pel nom del fitxer seguit de "_00.jpg" en cas que aquesta imatge sigui la imatge principal del producte (la que es veurà en la fitxa de producte més gran, en les categories i en la cistella de la compra) i "_01.jpg", "_02.jpg", etc seran les que es veuran a la galeria de fotos del producte, on només s'hi pot accedir des de la fitxa de producte. En cas que el client vulgui alguna altra distribució que no està contemplada (per exemple, que la imatge que es mostra en la categoria no sigui la principal) el programa ja té aquestes opcions però s'han d'especificar prèviament ja que estan deshabilitades, a part, s'haurà d'especificar el codi que caldrà posar en el nom de la fotografia per aquestes noves opcions. Aquest nom es pactarà amb el client i serà ell qui escollirà el prefix que posarà a les fotografies per la nova l'opció.

Una característica que cal remarcar d'aquest programa és que, en la importació d'imatges d'una botiga es va detectar que en els casos que la sku és numèrica, Magento pot confondre la sku per la id del producte i importar la imatge en un producte que no li correspon. Per evitar aquest problema es va fer que el programa busques prèviament la id del producte en funció de la sku i importar les imatges a través de la id. L'usuari final no té constància d'aquest procés ja que es fa tot internament i pot continuar utilitzant l'sku.

- **GoogleImportUpSell.php:**

Aquest controlador és l'encarregat de relacionar productes entre ells amb la finalitat de crear un efecte de "suggeriment de vendes". Magento té l'opció d'enllaçar productes entre ells perquè en alguns llocs de la botiga es puguin veure altres productes que el client ha relacionat prèviament. Aquest efecte pot ser que, quan estàs dintre de la cistella de la compra, depenent quins productes s'han afegit, la botiga "relaciona" uns productes i el client final es pot animar a comprar un altre producte. Els anomenats "up sells" són els que es veuen quan s'entra a la fitxa d'un producte sota la seva descripció, fotos, etc. Allà es veu un llistat de productes que són els que s'hauran assignat a aquest controlador. El client ha de crear una llista de skus de tots els productes que vol relacionar, separats per comes, a la columna de "ventas sugeridas". Un cop tenim aquesta relació sku (producte principal) -> llista d'skus (productes relacionats), el programa enllaçarà aquest productes i es podran veure a la botiga. S'ha implementat només aquesta opció ja que és la més utilitzada per part dels clients però, en cas de voler les altres opcions de Magento com ara "cross sell", es pot utilitzar el mateix controlador per fer l'assignació.

- **GoogleStockList.php:**

Aquest controlador és l'encarregat de fer un llistat d'estoc a Google perquè el client pugui veure en tot moment quin és l'estoc actual dels productes. El programa funciona de la següent manera:

El programa agafa primer el contingut del document a Google per saber quins productes estan ja llistats. Si es varia l'estoc, actualitza la cel·la corresponent. En cas que el producte no existeixi en el llistat, el programa introdueix el producte a la última línia. A més, el programa ha estat dissenyat de forma que no només mostra l'estoc del producte si no que és capaç de mostrar més informació del producte com ara el nom, marca del producte, etc. Els atributs que es volen mostrar en aquest llistat configuren un diccionari amb la mateixa estructura i tipus que utilitza el GoogleImport.php.

Helper: Són els encarregats de donar funcions de suport al programa. A continuació es llistaran els *helpers* que disposa el programa:

- **log.php i makeDictionaryGoogleImport.php:**

El *helper* log.php és una classe encarregada de generar un fitxer de log i, el makeDictionaryGoogleImport.php és l'encarregat de llegir un .csv (anomenat diccionari en el present treball, on hi ha la relació columna – atribut de Magento).

Els *helpers* que s'han esmentat estan dissenyats amb el patró *Singleton*, fent que si una classe crida aquesta, es crearà una instància i, si una altra classe que es fa servir en una

altre part del programa necessita d'aquesta primera, no es crearà de nou sinó que utilitzarà la mateixa instància que utilitza l'anterior classe. Això pot ser molt útil pel fitxer de log ja que, si no s'especifica el contrari, totes les classes guardaran els seus informes en el mateix fitxer ja que utilitzen la mateixa instància de classe on, en la seva inicialització, s'especifica el fitxer.

- `compareArraysGoogleImport.php` i `validateGoogleImport.php`:

Aquest dos *helpers* actualment no s'han dissenyat amb el patró *Singleton*: les classes `compareArraysGoogleImport.php` i `validateGoogleImport.php` són classes que el seu objectiu és comparar dos elements o validar si una dada és correcta, o no. Tot i que el seu objectiu és molt bàsic, al ser classes molt específiques que s'han dissenyat expressament per manipular les dades del importador, s'ha considerat que aquestes classes actualment no seran *Singleton* però, en un futur, poden ser filles d'una classe més abstracta anomenada *compare i validate* que aquesta funció "pare" faci funcions molt bàsiques de comprar dades o validar dades.

Lib: En aquest directori és on es guarden les parts necessàries del Zend Framework perquè funcioni el projecte. Totes les interaccions amb Google Docs estan implementades amb Zend Framework, per tant, es necessari tenir aquestes classes per poder interactuar, descarregar i manipular les seves dades.

Model: Aquí hi ha tots els models del projecte. Són els encarregats de manipular, guardar, i obtenir les dades. A continuació s'explicaran les característiques principals de cada classe:

- `dataObject`, `MClientSoap` i `MClientMagento`:

Aquestes classes són les encarregades d'obtenir la informació dels productes a Magento. El `dataObject` és la classe que criden els controladors per demanar un llistat de comandes, un llistat de productes, actualitzar un producte, etc. La característica que té `dataObject` és que, depenent de quin mètode de connexió s'especifica al crear la classe, s'obtidran les dades amb un mètode o amb un altre.

Per defecte, `dataObject` agafa la connexió SOAP ja que és la primera que es va implementar però, al veure que les connexions SOAP poden fallar i que el rendiment es veu afectat, s'ha començat a crear el `MClientMagento` per poder obtenir les dades a través de funcions que manipulen la base de dades i els models de dades de Magento directament. Tot i això, la classe `MClientMagento` encara no té totes les funcionalitats que té `MClientSoap` ja que s'han d'implementar totes les funcions des de zero i, en canvi, les funcions SOAP ja venen implementades a Magento.

- `GoogleClient` i `SpreadSheetClient`:

Aquestes dos classes són les encarregades d'obtenir la informació o manipular el contingut del full de càlcul de Google Docs. El `GoogleClient` està pensat en la creació de documents (en un futur es podria fer l'eliminació de documents) per inicialitzar el programa i, el `spreadSheetClient` està pensat per a la manipulació del document, introduint i agafant les dades que ha introduït el client.

- `imprtContent`:

Aquesta classe és el model de dades que utilitza el `SpreadSheetClient` per guardar la informació i poder ser tractada d'una manera més còmoda per part del controlador, el

comparador i el validador. Si en un futur es replanteja tenir una base de dades per guardar informació d'aquest programa, seria aquesta classe la que definiria els camps a guardar. El motiu de crear aquesta classe i no fer servir les dades tal com s'obtenen de Google és que s'ha considerat necessari poder manipular en camps separats l'sku del producte que es tracta, el llenguatge que s'ha introduït (en cas de que en tingui), el tipus de conjunt d'atributs del producte, etc. Així podem accedir a aquesta informació, útil i necessària per al programa, d'una manera més fàcil, còmoda i ràpida.

index.php: Aquest fitxer és el controlador principal que executa un controlador (especificat en la crida del programa) i carrega les variables de configuració que estan desades en els fitxers .ini. Aquest controlador fa la funció del patró *front controller*, fent que totes les crides a tots els controladors passin per ell i s'encarrega de gestionar aquestes crides.

Per cridar executar el programa, cal cridar el controlador index.php passant primer, el controlador que es vol executar i, com a segon paràmetre, la direcció del fitxer .ini. En el nostre cas, per executar el controlador principal GoogleImport.php, la crida en consola seria la següent:

- php -f index.php GoogleImport.php sites/pfc/pfc.ini
on el GoogleImport.php és el nom del fitxer on està el controlador i sites/pfc/pfc.ini és la ruta on està el fitxer pfc.ini

Idiomes, què implica?

A la primera versió del projecte no es va implementar l'opció dels idiomes. L'empresa va donar més prioritat a tenir el prototip funcional del projecte abans de poder tenir el suport per poder importar productes en més d'un idioma (reflectit en la planificació posant primer la creació de productes en front de l'actualització en diferents idiomes). A més, l'exigència real de poder importar productes multi idioma no va ser necessària fins al moment d'importar productes a la botiga www.runnering.com que té vistes de botiga tant en castellà com en italià.

Es va fer un replantejament de la lògica del sistema degut a la incorporació dels idiomes i de com s'ha d'importar la informació dels productes. Pel que fa a l'estructura de Google Docs va comportar afegir una columna anomenada "language" o "idioma" on, el client, introduiria el codi del llenguatge (es, en, it, etc). Un altre fet representatiu va ser que, de cara a l'usuari es va donar l'opció a poder tenir una sku repetida en el Google Docs. El plantejament inicial era que no es podia tenir skus repetides en el Google Docs i, si es detectava alguna sku repetida, s'informava a l'usuari de l'error ja que es podia perdre informació. Ara amb els idiomes no és possible aquesta comprovació ja que cada fila on es repeteixi l'sku indica que s'ha d'actualitzar el producte per un altre idioma.

Llista de comandes

Les llistes de comandes són llistats de les comandes que s'han realitzat a la botiga online perquè el client pugui tenir un millor control de la comptabilitat de la botiga. El programa que genera aquestes llistes no està inclòs en aquest programa i està en fitxers separats. El motiu principal és que els clients que han demanat la utilització de la llista de comandes en Google Docs han volgut unes llistes molt personalitzades: Alguns clients volien un resum de les comandes i altres volien un detall més elaborat amb informació específica de cada producte (com per exemple, la marca). Això ha fet que ara mateix el programa per llistar comandes no s'hagi desenvolupat en la seva totalitat i ens haguem centrat en crear un model base en què, depenent de les peticions del client, es personalitza d'una manera molt fàcil i ràpida.

El programa està format per un sol fitxer anomenat `orderListMagento.php` que obté les factures/factures d'abonament depenent d'una data determinada. L'execució del programa es realitza de la següent manera: Google crida una url cada dia a través d'un cron i envia aquesta data al programa. Aquest l'agafarà i mostrarà totes les comandes que s'han realitzat a partir d'aquesta data fins al moment actual. El motiu del control del cron per part de Google i no el nostre servidor és que el programa té l'opció de ser executat pel propi client des de Google Docs.

Per instal·lar el programa, s'ha d'inserir el codi d'execució a l'editor d'scripts de Google Docs i s'ha de posar el codi principal del programa en un directori visible des de l'exterior per poder ser executat al fer la crida a través de la url. Normalment el codi es diposita en la mateixa carpeta arrel que hi ha Magento.

Detalle de ventas y abonos ☆

Archivo Editar Ver Insertar Formato Datos Herramientas Ayuda eTailers-Actions La última modificación la ha realizado info hace 9 horas

Update Data

tipo	A	B	C	D	E	F	G	H
tipo	datafactura	idfactura	idpedido	fechapedido	nombre	sku	producto	
venta	1/12/2011 12:23:23	1201144435	2201144677	1/12/2011 12:21:36	ISABEL VIGHI BLANCO			
venta	1/12/2011 12:23:23	1201144435	2201144677	1/12/2011 12:21:36		MA13	Polvorones y mantecados artesanal	
venta	1/12/2011 13:05:55	1201144436	2201144678	1/12/2011 13:04:39	Cristina Oguiza			
venta	1/12/2011 13:05:55	1201144436	2201144678	1/12/2011 13:04:39		CMU11	Paella Typical Spanish	
venta	1/12/2011 13:05:55	1201144436	2201144678	1/12/2011 13:04:39		CMU14	Fideuá	
venta	1/12/2011 13:05:55	1201144436	2201144678	1/12/2011 13:04:39		445	Paletilla de lechazo asada al horno	
venta	1/12/2011 13:05:55	1201144436	2201144678	1/12/2011 13:04:39		456	Longaniza extra	
venta	1/12/2011 13:05:55	1201144436	2201144678	1/12/2011 13:04:39		1437	Morcón Ibérico puro de bellota	
venta	1/12/2011 13:05:55	1201144436	2201144678	1/12/2011 13:04:39		514	Higos confitados en almíbar	

Figura 19: Exemple de llistat de comandes a Google Docs.

En la figura 19, es pot veure el llistat exposat que conté les dades de la venda i els productes associats. Hi ha clients que no desitgen els productes en els informes, ja que els informes només els fan servir per a la comptabilitat. Tot això requereix una gran personalització del sistema.

Pel que fa al botó eTailers-Actions (figura 20) de la imatge serveix per a que el programa agafi la data que ha introduït el client en una cel·la i el programa mostri les noves comandes al final del document, quan se li demani.

	T	U	
	Desde_fecha	21-10-2011	

Figura 20: Cel·la on el client pot demanar les comandes des d'aquesta data fins el dia actual

Capítol 4: Proves i Aplicació del projecte

En aquest capítol citarem algunes botigues online on s'ha aplicat el projecte, donant dades reals i exemples de diferents importacions que han utilitzat aquest projecte. Les botigues seleccionades s'han citat perquè cadascuna té alguna característica especial que es vol destacar alhora de fer la importació. Per acabar, totes les proves i importacions que es citaran en aqu es van realitzar amb èxit i són importacions en entorns reals.

Vila viniteca

La web www.vilaviniteca.com/shop és una botiga dedicada a la venda de vins i begudes alcohòliques, a més d'altres productes. Aquesta botiga es caracteritza per tenir molts atributs únics per les característiques dels productes. Per als vins, es van haver d'importar atributs com l'anyada, les varietats, els graus d'alcohol, el tipus de vi, etc. Donat que la web ofereix un ampli ventall de productes a més dels vins, que és la seva especialitat, com llibres, pernills i altres productes que tenen altres característiques es van haver de crear varis conjunts d'atributs. Cada conjunt té diferents atributs i diferents característiques (autor del llibre, zona, marca, pes del pernil...). Per poder importar tots aquests productes, es van crear varis fulls de càlcul Google Docs (figura 26) on a cadascun es van introduir tots els productes de cada tipus (figura 27). Aquesta disposició va ajudar a poder localitzar els tipus de productes fàcilment i, a més, al ser un projecte on molts clients introduïen productes, la utilització de Google docs va ser perfecta per a la situació.

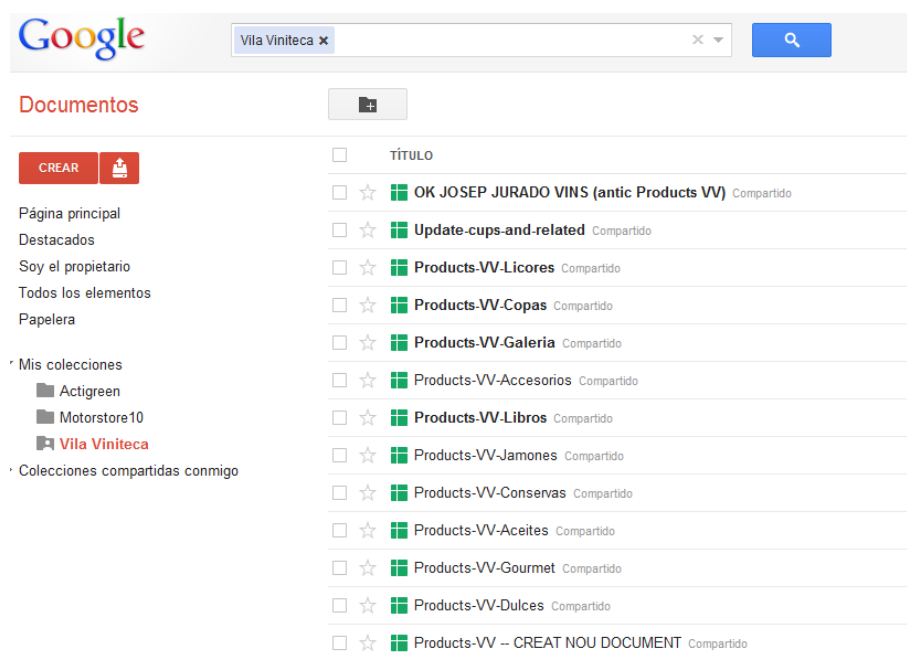


Figura 26: Llistat de Google Docs que es van utilitzar per importar tots els productes de vilaviniteca, en total es van importar més de 3000 productes simples.

OK JOSEP JURADO VINS (antic Products VV) ☆

Archivo Editar Ver Insertar Formato Datos Herramientas Ayuda Se han guardado todos los cambios.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	sku	tipo-de-conj	tipo	tienda	short-descri	description	bodega	status	tamano	color	variedades	variedades2	variedades3	variedades4	pa
1557	029030	Vino	simple	base			Bodegas San Alejandro	Habilitado	75 cl	Tinto	Garnacha				Es
1558	029040	Vino	simple	base			Mondavi	Habilitado	75 cl	Tinto	Cabernet Sa	Petit Syrah	Syrah	Otras varied	Es
1559	029042	Vino	simple	base			St. Urbans-Hof	Habilitado	75 cl	Blanco	Riesling				Alk
1560	029043	Vino	simple	base			Château Fuissé	Habilitado	75 cl	Blanco	Chardonnay				Fr
1561	029044	Vino	simple	base			Château Fuissé	Habilitado	75 cl	Blanco	Chardonnay				Fr
1562	029050	Vino	simple	base			Portal del Montsant	Habilitado	1,5 l	Tinto	Carifena	Syrah	Garnacha		Es
1563	029051	Vino	simple	base			Bodegas Mauro	Habilitado	3 l	Tinto	Tempranillo	Syrah	Garnacha		Es
1564	029052	Vino	simple	base			Bodegas Mauro	Habilitado	6 l	Tinto	Tempranillo	Syrah	Garnacha		Es
1565	029056	Vino	simple	base			Portal del Priorat	Habilitado	75 cl	Tinto	Carifena	Syrah	Garnacha	Otras varied	Es
1566	029059	Vino	simple	base			Portal del Priorat	Habilitado	5 l	Tinto	Carifena	Syrah	Garnacha	Otras varied	Es
1567	029067	Vino	simple	base			Dönnhoff	Habilitado	75 cl	Blanco	Riesling				Alk
1568	029068	Vino	simple	base			Dönnhoff	Habilitado	75 cl	Blanco	Riesling				Alk
1569	029069	Vino	simple	base			Ca N'Estruc	Habilitado	75 cl	Tinto	Syrah	Carifena	Garnacha		Es
1570	029074	Vino	simple	base			Ökonomierat Rebholz	Habilitado	75 cl	Blanco	Riesling	Spätburgun	Weissburgu	Otras varied	Alk
1571	029077	Vino	simple	base			Ökonomierat Rebholz	Habilitado	75 cl	Blanco	Gewürztram				Alk
1572	029086	Vino	simple	base			Bott-Geyl	Habilitado	75 cl	Blanco	Pinot Blanc	Pinot Auxer	Pinot Gris	Pinot Noir	Fr

Figura 27: Exemple de llistat de vins d'un full de càlcul de vila viniteca.

YouBuyOutdoor

La web <http://www.youbuyoutdoor.com/oues/> ofereix productes per a la muntanya. Estan especialitzats en l'escalada, acampades, trekking... La característica especial d'aquesta botiga és que, tot i que de cara a l'usuari final és una botiga única, en realitat és una botiga que utilitza la mateixa base de dades de www.runnering.com. Això és degut a que els propietaris de runnering i youbuyoutdoor són els mateixos i es va demanar poder gestionar les dues botigues des del mateix backoffice de Magento. Això va fer importar els productes només a la botiga youbuyoutdoor tot i que comparteix la mateixa base de dades amb runnering. Per fer-ho, es va haver de canviar el valor del camp "website" (en Google pot tenir un altre nom degut al diccionari que es fa servir) de Google Docs a "ou" en lloc de "base" com és habitual (figura 29). A més, en aquesta botiga també s'ha requerit la importació de productes configurables (figura 28). En aquesta botiga s'han importat més de 1000 productes.

NOVEDADES

MÁS VENDIDOS

OFERTAS

OUTLET


Newsletter

email

Suscríbete

Acepto la [Política de privacidad](#)

Zapatillas Salomon XA PRO 3D ULTRA DEEPRED



Te recomendamos las Botas Salomon XA PRO 3D ULTRA DEEPRED. Estas botas son el mejor calzado de trail running gracias a su nueva amortiguación. Para llevarte del camino a la montaña con una estabilidad sin precedentes en cualquier temporada del año y sobre todos los terrenos.

Disponibilidad: En existencia

119,90€

Opiniones y preguntas sobre este producto

0 Me gusta 2 Twitter

*Talla

Seleccione una opción...

Seleccione una opción...

EU 41

EU 42

EU 42.5

EU 43

EU 44

EU 44.5

Dejanos un comentario

Zapatillas Salomon XA PRO 3D ULTRA DEEPRED Descripción

Conoce las botas Salomon XA PRO 3D ULTRA DEEPRED y sus prestaciones:

Con membrana Core-Tex® Extended Comfort Footwear: mantiene los pies secos y protegidos por ser duradera, impermeable y transpirable.

Figura 28: Producte configurable importat a youbuyoutdoor amb el sistema Google Docs, en la imatge podem veure com es pot seleccionar la talla del producte.

Yobuyoutdoor-Productes ☆

Archivo Editar Ver Insertar Formato Datos Herramientas Ayuda Se han guardado todos los cambios.

fx sku											
	A	B	C	D	E	F	G	H	I	J	K
1	sku	tipo-de-conjunto-de-atributo-config	tipo	categoria:	tienda	nombre	estado	tipo-de-im	cost	precio	prec
2	Ybo_O20	Pantalones / camisetas (XL-S)	Configura	810, 840,	ou	Medias d	Habilitad	IVA 18%	17.32		37
3	Ybo_O20	Pantalones / camisetas (XL-S)	simple	810, 840,	ou	Medias d	Habilitad	IVA 18%	17.32		37
4	Ybo_O20	Pantalones / camisetas (XL-S)	simple	810, 840,	ou	Medias d	Habilitad	IVA 18%	17.32		37

Figura 29: Full de càlcul de Google Docs de yobuyoutdoor, podem veure a la imatge com la columna tienda té el codi "ou" en lloc del codi "base"

Motorstoreshop

www.motorstoreshop.es és una botiga especialitzada en articles de motos. Tot i que la botiga està pendent d'un redisseny que serà llançat pel febrer de 2012, s'han hagut d'importar una gran quantitat de productes configurables i, a més, amb més d'un atribut configurable per producte com es pot veure a la figura 30.

Shop Categories

- Homepage
- Accesorios motorista
- Cascos Moto
- Cascos Pilotos
- Guantes moto
- Marcas
- Chaquetas moto
- Pantalón moto
- Botas moto
- Motos eléctricas
- Merchandising pilotos
- Jorge Lorenzo
- Ferrari
- Ducati
- Blog

Inicio > Guantes moto Garibaldi - Rookie



Doble click en la imagen para abrir/cerrar

MÁS VISTAS




Guantes moto Garibaldi - Rookie

Disponibilidad: En existencia

Precio regular: €29,95

PRECIO ESPECIAL: € 25,38

+1 0 Me gusta

Descripción rápida

Los guantes moto Rookie de la marca Garibaldi son una pieza de vestir imprescindible apr acunado vayas en moto. Gracias a su especial fibra los guantes moto Rookie son un componente clave para tu seguridad.

***Color**

Seleccione una opción...

***Talla**

Seleccione una opción...

* Campos requeridos

Precio regular: €29,95

PRECIO ESPECIAL: € 25,38

Cantidad: 1

AGREGAR AL CARRITO

Figura 30: Producte de motorstoreshop. Podem veure en la imatge com en aquest producte s'ha de seleccionar tant el color del guant com la seva talla.

Archivo Editar Ver Insertar Formato Datos Herramientas Ayuda Se han guardado todos los cambios

Figura 31: Part del Google Docs de Motorstoreshop.

Selegto

En la figura 33, on es pot visualitzar el llistat d'estoc, destaquem que el client volia visualitzar més informació del producte que una simple relació sku – estoc. Per aquest motiu es visualitzen més columnes com ara el nom, gestió d'estoc, pes, etc.

[Archivo](#)
[Editar](#)
[Ver](#)
[Insertar](#)
[Formato](#)
[Datos](#)
[Herramientas](#)
[Ayuda](#)
[eTailers-Actions](#)
 La última modificación la ha realizado info hace 12 horas

Figura 32: Part del document del llistat de vendes i factures d'abonament de selegto.com

Listado_Stocks_Selegto ☆

Archivo Editar Ver Insertar Formato Datos Herramientas Ayuda eTailers-Actions La última modificación la ha realizado info hace 100 días

										Most
A		B		C		D	E	F	G	H
	SKU	Type	Name	qty	gestion-de-stock	Deposito	Peso	marca		
	273	simple	Huevas de Trucha "Eris Mar"	9	Si	No	0,1	Eris Mar		
	274	simple	Huevas de Salmón salvaje "Eris Mar"	10	Si	No	0,1	Eris Mar		
	275	simple	Tartar de Algas "Eris Mar"	9	Si	No	0,09	Eris Mar		
	276	simple	Trinxat de Algas "Eris Mar"	10	Si	No	0,018	Eris Mar		
	277	simple	Judías de Mar "Eris Mar"	10	Si	No	0,2	Eris Mar		
	278	simple	Anchoa del Cantábrico 00 "Arlequin"	40	Si	No	0,25	Arlequin		
	279	simple	Berberechos de las Rias Gallegas "Arlequin"	86	Si	No	0,125	Arlequin		
	268	simple	Aceite de oliva v e., 200 ml. "AME Touch"	6	Si	No	0,2	AME Touch		
	280	simple	Vino tinto reserva 2006 Abadia Mediterránea "Abadia Mediterránea"	11	Si	No	0,75	Abadia Mediterranea		
	100000	simple	Aceite de Oliva V.E. "Buxeda Moli d'Oli"	494	Si	No	0,5	Buxeda Moli d'Oli		
	281	simple	Vino tinto crianza 2007 Abadia Mediterránea "Abadia Mediterránea"	11	Si	No	0,75	Abadia Mediterranea		
	282	simple	Vino tinto reserva 2005 Templar "Templar"	11	Si	No	0,75	Templar		
	283	simple	Vino tinto crianza 2007 Templar "Templar"	11	Si	No	0,75	Templar		
	285	simple	Vino tinto reserva 2001 Cossetania "Cossetania"	10	Si	No	0,75	Cossetania		
	286	simple	Vino tinto 2006 Cossetania "Cossetania"	12	Si	No	0,75	Cossetania		

Figura 33: Part del document de llistat de productes de seleqto.

Altres botigues

Per il·lustrar més exemples de webs on s'ha aplicat el projecte, convidem a visitar el següent llistat:

- www.accessoriosdecoches.com: venda de productes i accessoris per el cotxe.
- www.actigreen.com: vende de productes ecològics i sostenibles.
- www.canadahouse.es: venda de roba per nens i nenes.
- www.leche-infantil.com: venda de llets i productes per infants.
- www.runnering.com: venda de productes d'esport i per corredors,
- www.sololimpieza.com: venda de productes de neteja de la marca vileda.
- www.vivabebe.com: venda de productes per infants.

La diferència entre aquestes botigues i les llistades anteriorment és que no tenen cap característica especial que cal destacar i per això s'han mostrat sense entrar en profunditat. Les seves importacions es van realitzar amb èxit i sense cap incidència.

Capítol 5: Conclusions

En aquest capítol es revisaran els resultats del projecte: objectius assolits, intencions de futur, valoració personal i es comentarà la utilització del programa per part dels clients i els seus comentaris. També es tornarà a fer una revisió de la planificació ja que hi han hagut diferents canvis que han fet variar el que s'havia establert inicialment, mostrant la planificació final del projecte.

Objectius assolits

Un cop finalitzat el desenvolupament del projecte cal analitzar de manera exhaustiva si s'han complert els objectius proposats en el Capítol 1. En aquest apartat s'havien anomenat els diferents objectius que l'empresa demanava per solucionar diferents problemes en què es trobava. Revisem-los un a un:

Simplificar el sistema d'importació de productes, fent que molts dels atributs que necessita Magento per importar un producte s'emplenin automàticament.

Considerem aquest objectiu assolit. Si es consideren només els atributs obligatoris, s'ha passat de tractar més de 70 a tractar-ne menys de 10.

Fer ús de l'API de Google Docs per, en lloc d'introduir els productes des d'un fitxer csv, s'introdueixin des d'un full de càlcul Google docs.

Poder importar els productes des d'un full de càlcul Google Docs a una botiga Magento.

El projecte utilitza l'API de Google per poder descarregar la informació d'un full de càlcul de Google Docs i poder importar els productes. Es considera que aquest objectiu s'ha assolit.

Importar diferents productes en diferents idiomes:

Els productes es poden importar en diferents botigues i en diferents idiomes.

Importar imatges dels productes introduïts al Google Docs:

S'ha creat un controlador extern que permet importar imatges a Magento.

Si els productes ja existeixen a la botiga on-line, el programa ha de ser capaç de identificar-los i actualitzar el seu contingut amb el que s'ha introduït a Google:

Es permet actualitzar els productes que ja estan creats a Magento.

Que el programa pugui validar els valors que introdueix el client a Google.

El programa valida si els valors que l'usuari ha introduït són correctes.

Estudiar el llenguatge Python i l'API de Google docs per a la realització de la importació:

Tot i que finalment el programa s'ha reprogramat en PHP, es va poder desenvolupar una gran part amb Python i, amb aquest desenvolupament, el programador va poder estudiar i familiaritzar-se amb el llenguatge.

Estudiar la compatibilitat entre PHP i Python per a la interacció entre Magento i el programa:

Es va estudiar la interacció i no es va veure factible. Aquest és el motiu principal de la reprogramació del sistema.

Realitzar un parser per poder fer un estudi de mercat de tots els productes que hi ha a Google Docs, fent una comparativa de preus.

Aquest objectiu no s'ha assolit. L'empresa va decidir donar més prioritat a altres apartats i afegir altres funcionalitats al sistema que dedicar esforços i recursos a aquest objectiu.

Així doncs, pel que fa als objectius més prioritaris, s'han assolit tots. L'objectiu amb menys pes que tenia el projecte és l'únic que no s'ha assolit però, com s'ha explicat en la memòria, en lloc de centrar esforços en el comparador s'ha destinat el temps i recursos a crear l'informe d'estoc o l'informe de vendes.

Línies d'ampliació

Tot i que aquest projecte hagi assolit una versió estable i que compleix tots els requeriments, creiem convenient estudiar les possibles vies o línies que ha de seguir el programa per continuar endavant i admetre la possibilitat d'actualització:

- Reescriptura de les connexions SOAP a funcions internes de Magento: Aquesta part ja s'està fent en l'actualitat tot i que no s'ha especificat en aquesta memòria. En la memòria s'ha tractat l'evolució del projecte fins a tenir un model estable i funcional però, per millorar el rendiment i l'estabilitat del sistema, actualment s'està reescrivint el codi de manipulació de la base de dades Magento de SOAP a funcions de Magento.

La reescriptura de les connexions està enfocat a millorar el rendiment del sistema. El programa funciona perfectament per importar productes però, si s'ha de programar un cron per botigues que tenen molts productes (per exemple, www.runnering.com) l'execució del programa és molt costosa degut a que ha de processar les dades de Google, les de Magento i fer les importacions/actualitzacions corresponents. A més, també poden sorgir dificultats al crear una connexió SOAP (sessions que caduquen, error en l'enviament en les dades, etc).

Un cop s'hagi reescrit el codi referent a les connexions SOAP, es podran crear crons per programar periòdicament l'execució del programa per botigues més grans i amb més productes. Actualment, només es recomana l'execució del programa en botigues amb pocs productes. Això compleix el propòsit inicial del nou model de negoci que havia plantejat l'empresa.

- Integració del projecte a Magento: Hi ha la possibilitat d'integrar el projecte completament a Magento, creant per a ell un mòdul i poder publicar-lo en el Magento connect. El rendiment no es veurà afectat però si la seva integració en la plataforma.
- Actualització de l'estoc: Tot i que ara ja tenim un programa que fa un llistat de tots els estocs que té la web, encara no està implementada l'opció d'actualitzar l'estoc. S'ha de desenvolupar aquesta part.
- Manipulació de comandes: Actualment el llistat de comandes està desenvolupat però encara no s'ha implementat la seva manipulació. El programa hauria de treballar aquesta tasca per poder enllestir el nou model de negoci que vol tenir l'empresa.

Valoració personal

El desenvolupament del projecte m'ha semblat una experiència enriquidora en molts sentits. He pogut utilitzar coneixements que he après durant el transcurs dels estudis universitaris i aplicar-los al desenvolupament d'un projecte real construït des de zero. A més, gràcies a alguns problemes que han sorgit durant el desenvolupament m'ha calgut consultar fonts d'informació externes per aprofundir més en algunes àrees de les que tenia coneixements molt superficials.

Crec que gràcies a aquest projecte he pogut millorar considerablement els meus coneixements en les tecnologies PHP, Magento, etc, i he pogut obtenir coneixements amb un nou llenguatge de programació que no coneixia com Python. Aquests coneixements ja els he pogut aplicar a altres projectes que he hagut de fer per a l'empresa i considero que els meus coneixements informàtics han madurat amb aquests projecte.

A més, he après a gestionar i documentar correctament un projecte gran, cosa que em pot ser molt útil amb la meua carrera professional.

Per acabar, considero que la valoració personal que tinc del projecte és molt positiva i, a més, puc veure com el meu projecte s'aplica a una empresa real i es fa servir durant el dia a dia de l'empresa.

Utilització per part dels clients

La primera pàgina web que es va crear utilitzant el nou sistema d'importació va ser www.vilaviniteca.com. Es va escollir aquesta web degut a que ells mateixos van reconèixer que hi havia molta gent implicada en la introducció d'informació dels productes a la botiga. En utilitzar Google Docs, totes aquestes persones van poder treballar en paral·lel, en el mateix document i en temps real. L'experiència per part del client va ser positiva i van trobar el sistema molt fàcil d'utilitzar ja que només havien de posar d'informació necessària per crear el producte.

En altres projectes es va trobar la situació que algun usuari havia esborrat informació que no s'havia d'esborrar. La recuperació d'informació va ser molt ràpida i fàcil degut a que Google manté un historial de tots els canvis que s'han produït en el document.

Hi ha hagut clients que no han trobat "ideal" el fet de treballar directament el document a Google Docs. Per aquests casos el que s'ha fet és que ells es descarreguin el document en format Excel, treballin localment i, quan ells ho decideixin, s'ha importat el document a Google. Aquests casos han estat pocs i, a més, han estat projectes on només treballava una sola persona carregant la informació dels productes.

Pel que fa a l'empresa, també està utilitzant el programa en el departament de Marketing online. Aquests han arribat a varis acords amb diferents botigues per importar productes un cop per setmana. Aquestes botigues són www.runnering.com, www.youbuyoutdoor.com, www.accesoriosdecoches.com, etc. A més, també s'està utilitzant per botigues que l'empresa en té els drets com ara www.motorstoreshop.com. L'acceptació d'aquest nou sistema ha estat molt gratificant per l'empresa ja que ha permès poder realitzar entre 2 i 3 importacions diàries on, abans es podia realitzar una al dia o una cada dos dies. Gràcies al sistema s'han pogut arribar a aquests acords ja que s'ha agilitzat el sistema d'importacions.

Millores respecte al sistema inicial

Com hem anat explicant al llarg de la memòria, l'anterior sistema requeria un fitxer on l'administrador havia de gestionar al voltant de 70 columnes. Amb aquests nou sistema les columnes obligatòries s'han reduït a 4 en cas de voler crear un producte o 7 en cas de voler crear productes configurables. La millora ha estat molt significativa ja que el nombre d'errors en les columnes disminueix dràsticament.

Una altra part on s'ha millorat molt és el temps dedicat a la importació amb la preparació del document. Abans, quan el client retornava el fitxer .csv amb els seus documents, l'administrador havia d'estar un temps (depenent de si el document tenia errors o no) en preparar-lo per introduir tots els atributs necessaris per importar. Ara, la validació del document la pot fer el mateix programa fent que, si s'executa el programa amb el flag "importMode" a "true", el mateix programa elabora un resum de les accions que farà i haurà validat el document a Google; notificant al client quins errors ha generat i, depenent dels errors, podent ser corregits pel mateix administrador o, si es necessari, amb la intervenció del client.

Reduint el temps per preparar el document i millorant el sistema de validació, s'han pogut agilitzar les importacions, planificant de una importació al dia a poder fer més de tres importacions diàries en botigues diferents. És millor no fer més importacions perquè primer es fa una importació de prova en un entorn local per comprovar que realment no hi hagi cap error que no hagi estat detectat pel programa.

Pel que fa a les validacions o errors que detecta el sistema també s'ha millorat dràsticament. Els missatges d'error que mostrava el sistema anterior no sempre eren prou clares i, a més, no es validaven els valors dels atributs configurables; cosa que podia fer repetir la importació diverses vegades degut a que els simples no tenien el valor que tocava. Ara el programa fa una validació dels camps més importants i, quan tenen valors incorrectes, poden ser corregits en una futura importació.

Planificació final

La segona planificació tampoc es va poder assolir degut a retards en el desenvolupament. Tot i que aquests retards s'expliquen amb més detall en el capítol d'implementació, ara es farà un llistat dels inconvenients que s'han anat trobant al llarg del desenvolupament:

1. Necessitat de desenvolupar funcionalitats que, inicialment, se suposava que ja estaven creades en Magento.
2. Introducció de nous projectes que, segons el criteri de l'empresa, eren de més prioritat i requerien els recursos destinats a aquest projecte.
3. Errors inicialment desconeguts que han suposat fer replantejaments del projecte.
4. Canvi de tecnologia (passar de Python a PHP).
5. Reescriptura de parts del codi.

Partint de la base que la memòria requeria un mes per redactar-la i supervisar-la, es va decidir ajornar el projecte fins febrer. Durant aquest temps es va poder acabar en gran part el projecte i s'ha pogut utilitzar en entorns reals. Això ha premès fer proves i obtenir un feedback molt útil per a la millora del projecte.

La planificació final del projecte, adjuntant les noves tasques, retards i problemes que hi ha hagut, ha resultat tenir el llistat de tasques mostrat a la figura 34 i el diagrama de grantt de la figura 35:

El cost per assolir aquestes tasques està calculat per la següent manera:

- Cost per la llicència i utilització de Magento: 0 €
- Cost per la llicència i utilització de Google Docs: 0 €
- Hores de treball aplicades al projecte: s'ha estimat que s'assignaran al voltant de 550 h de treball al projecte. $550 \text{ h} \times 4 \text{ €/h} = \mathbf{2200 \text{ € de cost al projecte per part de l'empresa.}}$



  Task	Effort	Unique ID
▼ 1) Informe prèvi	2w 4d 1h	8
• 1.1) Definició d'objectius	0,5h	2
• 1.2) Definició de les parts	0,5h	9
• 1.3) Redactat del document	1w 3d	10
• 1.4) Revisió i supervisió	1w 1d	11
▼ 2) Anàlisis	6w	73
• 2.1) Anàlisis del problema	2w	74
• 2.2) Anàlisis del importador de Google	2w	75
• 2.3) Anàlisis del comparador	2w	76
• 3) Disseny de classes	1w	79
▼ 4) Desenvolupament	37w 3d	14
▼ 4.1) Google Docs	25w	16
• 4.1.1) Estudi de python i api de google docs	1w	15
• 4.1.2) Connexió i obtenció de documents google docs	1w	17
• 4.1.3) Manipulació del document	1w 3d	18
• 4.1.4) Reescriptura del projecte	2w	88
▼ 4.1.5) Enviament dels productes a la web	11w	71
▼ 4.1.5.1) Creació dels productes en la e-shop	6w	64
• 4.1.5.1.1) Creació de productes simples	2w	89
▼ 4.1.5.1.2) Creació de productes configurables	4w	90
• 4.1.5.1.2.1) Implementació d'un mòdul Magento	3w	91
• 4.1.5.1.2.2) Creació dels productes configurables en la e-shop	1w	93
• 4.1.5.2) Actualització dels productes en la e-shop	1w 3d	66
• 4.1.5.3) Actualització dels productes en diferents idiomes	4d	68
• 4.1.5.4) Validació de camps i correcció d'errors	2w 3d	19
• 4.1.6) Sistema de log	2w	21
• 4.1.7) Test i correcció d'errors	3w	22
▼ 4.1.8) Importació d'imatges	3w 2d	24
• 4.1.8.1) Estudi de com importar les imatges.	1d	25
<i>A pactar amb l'empresa el disseny</i>		
• 4.1.8.2) Implementació	2w	27
• 4.1.8.3) Test	2d	26
• 4.1.8.4) Millora del importador d'imatges	4d	96
▼ 4.2) Visualitzador de ventes Google Docs	3w 2d	28
• 4.2.1) Estudi d'obtenció i manipulació de les ventes	3d	29
• 4.2.2) Enviament de les ventes a Google	2d	30
• 4.2.3) Manipulació de les ventes	2w	31
• 4.2.4) Test	2d	33
▼ 4.3) Gestió d'stock	4w 1d	82
• 4.3.1) Estudi de la gestió d'stock amb Google docs	1w	83
• 4.3.2) Enviament de la informació a Google	3d	84
• 4.3.3) Actualització d'stock	2w	85
• 4.3.4) Test	3d	86
• 4.4) Instal·lació i feedback	5w	94
• 5) Realització de la memòria	45w 2d 3h	43
• 6) Correcció de memoria	20w	95

Figura 34: Tasques de la planificació final

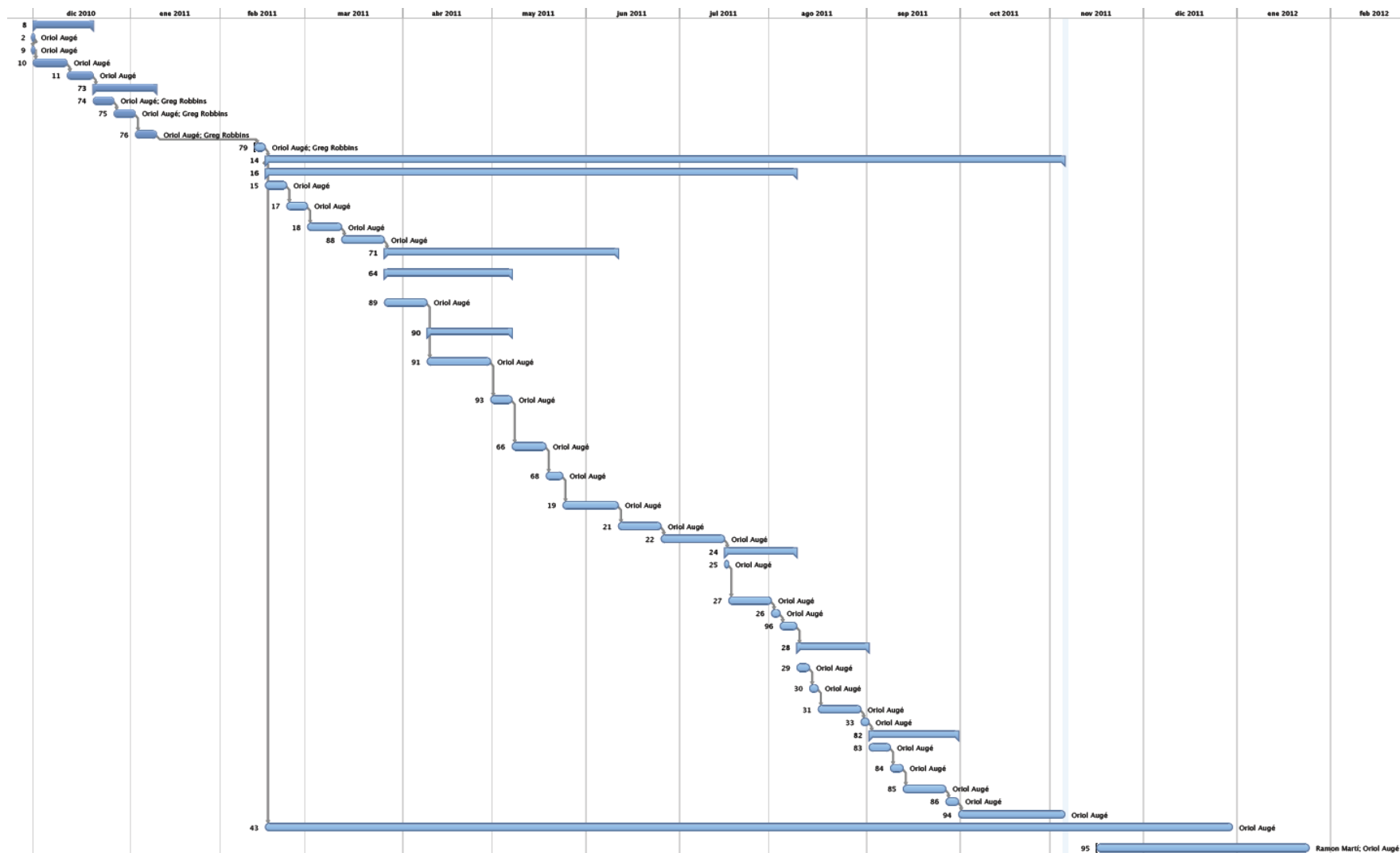


Figura 35: Diagrama de Grantt final. Cada tasca està identificada amb la seva Id.

Annex: Instal·lació del projecte

En aquest annex ens centrarem en analitzar el funcionament del projecte, fent d'aquesta part un manual d'instal·lació i utilització del programa.

L'explicació que es donarà estarà orientada al perfil de l'administrador del sistema, ja que el client final només haurà d'introduir la informació específica a cada columna de Google Docs. S'explicarà la instal·lació bàsica del programa, les característiques d'un full de càlcul que conté productes configurables i s'exposarà un exemple d'un missatge d'error que pot veure el usuari final al introduir dades.

Manual del programa: Instal·lació

Abans que el client pugui introduir les dades, a d'intervenir l'administrador per preparar l'entorn a Google. El procediment de creació és el següent:

1. **L'administrador crearà un .csv que contindrà les columnes necessàries per poder crear i/o actualitzar el document.** El disseny estàndard del document és el següent:

Magento	Google	Valor d'exemple
sku	sku	AD123
attribute_set	tipo-de-conjunto-de-atributo-configurable	Default
type	tipo	Simple
website	tienda	Base
visibility	visibilidad	Catálogo, Búsqueda
language	idioma	Es
categories	categorias	3,4
short_description	Descripcion-corta	Desc. Curta
description	Decripcion	Descripció llarga del producte
status	Estado	Habilitado
price	Precio	10
special_price	Precio-especial	5.6
weight	Peso	0.1
name	Nombre	Nombre
tax_class_id	Iva	IVA 18%
stock_data	Cantidad-en-stock	5

La primera columna representa el nom de l'atribut en Magento, la segona columna representa el nom de les variables que consten en el full de càlcul de Google Docs i la tercera columna, proporciona un exemple de contingut. Els noms que s'introdueixen en la columna central són exemples i, per tant, poden ser diferents als de la taula. L'única característica que s'ha de tenir en compte és que quan Google retorna el contingut del document, Google elimina les “_”, “ ” i

els accents, fent que els noms s'hagin de separar amb "-". Per visualitzar millor la relació entre el diccionari i el full de càlcul Google Docs, podem veure la figura 21:

sku	tipo-de-conjunto-de-atributo-configurable	tipo	tienda	categorias	short-description	description
132362	Default	simple	base	Bayetas	Bayeta amarilla multusos sin olores extra grande y extra gruesa.	•Multusos súper absorbente y suave •Dimensión gigante •Su mayor grosor aliado a una mezcla confieren una gran capacidad de seco •Ideal para la limpieza en seco o en m superficie.
133371	Default	simple	base	Bayetas	Bayeta amarilla multusos con tratamiento sin olor. Máxima absorción y limpieza. Lote 6 unidades.	•Multusos súper suave y manejable. G y recogida de suciedad. Tecnología d absorbente. •Hecha de fibras de alta c mayor absorción y una suavidad y res •La micro perforación garantiza una g de suciedad •Ideal para la limpieza en seco o en m superficie.
					Bayeta amarilla multusos	•Multusos súper suave y manejable. G y recogida de suciedad. Tecnología d absorbente. •Hecha de fibras de alta c mayor absorción y una suavidad y res •La micro perforación garantiza una g

Figura 21: Relació entre el diccionari guardat en el csv i el full de càlcul Google Docs

La *sku* sempre ha de ser el primer atribut del document. Això és així perquè quan es demana el contingut del document a Google (*getListFeed*), la primera columna es fa servir com a títol de la fila. Es va pactar amb l'empresa que el millor títol (o identificador) que es podia donar a cada fila era l'*sku*. L'*sku* és un camp obligatori.

attribute_set defineix el conjunt d'atributs del producte. Gràcies a aquest atribut podem validar una gran quantitat de valors introduïts al document i és necessari per a la creació de nous productes. Està considerat com un camp obligatori.

Els atributs *type*, *webiste* són atributs necessaris sempre i quan es pretengui crear productes a la botiga. Quan es crea un producte s'ha d'especificar a quina botiga ha d'anar (recordem que Magento suporta múltiples botigues amb una mateixa instal·lació) i cal saber quin tipus de producte és (si és simple o configurable). Aquests productes són només necessàries per a clients que volen introduir productes que no existeixen a la botiga ja que, si es el que es vol és actualitzar la informació d'un producte, no és necessari especificar aquesta informació. Això ho destaquem perquè ens hem trobat clients que només els interessa actualitzar preus massivament de productes que ja existeixen a la botiga. En aquests casos es crearà el csv només amb els atributs *sku*, *attribute_set* i *price*. També cal destacar que l'atribut *website* no és indispensable ja que si no està omplert aquest camp, el producte es crearà però no s'assignarà a cap botiga. Pot ser algun client consideri important aquest fet però, com no ens hem trobat aquesta situació i sempre que es vol importar un producte és per anar a una botiga determinada, considerem aquest camp com obligatori i així ho transmetem als clients.

Per saber quins valors s'han d'introduir al camp *website*, s'ha d'accedir al backoffice i analitzar el codi de la botiga. Per defecte, la botiga principal té el codi "base" però, si es vol introduir el producte en una sub-botiga, s'ha d'obtenir el codi d'ella per poder importar el producte. Per obtenir el codi, cal anar al backoffice de Magento, accedir a l'apartat Sistema → Gestionar tiendas (figura 22).



Figura 22: Backoffice de Magento. Apartat Sistema -> Gestionar Tiendas

Un cop en aquest apartat, s'accedeix al "Sitio web" on es vol importar els productes i un cop allà, s'obté el codi de la botiga (figura 23). Per més informació sobre els *websites* i l'estructura de botigues de Magento es recomana el manual [magento2] de la bibliografia.

Editar el sitio web

Información del sitio web

Nombre *	<input type="text" value="website.com"/>
Código *	<input type="text" value="base"/>
Ordenar pedido	<input type="text" value="0"/>
Tienda por defecto	<input type="text" value="Basic Eshop"/>

Figura 23: Website d'una configuració per defecte de Magento, podem veure el camp "Código" que necessitem per importar els productes a aquesta botiga. Aquesta informació sempre se li proporcionarà al client ja que no acostuma a tenir accés a aquest nivell del backoffice.

Els camps següents ja no són obligatoris i, encara que molts puguin ser molt importants (com, per exemple, l'iva que s'aplica al producte) la seva absència no impediria que es creés el producte a la botiga.

language és la columna on s'ha de indicar l'idioma del producte. En cas que la botiga no tingui idiomes, aquesta columna es pot eliminar.

La columna *status* també és un atribut obligatori per poder crear un producte però s'ha pactat amb l'empresa que és millor crear el producte i deixar-lo com deshabilitat.

Camp *categories* és el lloc on s'indica a quines categories es posarà el producte. Aquest camp es pot posar tant el nom de la categoria (Ex: hombre), la id de la categoria (Ex: 24) o intercalar les dues possibilitats (Ex: hombre, 43, 56).

tax_class_id és on s'ha d'indicar l'iva que s'aplica al producte (o la classe d'impostos si no és un producte espanyol). El nom que s'ha d'introduir aquí està fixat pel backoffice de Magento i s'ha hagut de crear prèviament. Si a Magento s'ha creat l'iva "IVA 18%", serà aquest el valor el que s'haurà d'introduir en aquests casos.

Csv en els productes configurables

En els productes configurables s'han d'afegir les següents columnes especials:

Magento	Google	Valor d'exemple
<i>associated</i>	Productos-simples-asociados	AD3,AD4
<i>config_attribute</i>	atributo-configurable	talla

La columna *associated* indica els productes simples que s'associaran al configurable, és un atribut que només ha de tenir omplert el configurable i, en cas dels simples, s'ignorarà.

config_attribute indica, tant als simples com als configurables, quin és l'atribut a què s'associaran. Un cop s'ha indicat l'atribut, s'ha de crear una altra columna per cada atribut per indicar el valor que tindran els simples en relació a l'atribut. Seguint l'exemple que apareix a la taula, s'hauria de crear una columna anomenada talla on els simples tindrien el valor que correspon a cadascun d'ells.

2. Un cop creat el csv. **L'administrador ha de configurar el fitxer .ini que conté les dades d'accés al Google docs, csv i connexió SOAP.** La plantilla ja preparada per el .ini és la següent:

```
[google_docs]
email = "email@gmail.com"
pass = "*****"
source = "Etailers-GoogleImport-v1.5"
fn = "Products-eShop0"
dictionary= "<path_of_your_dictoinary>.csv"
timeOut = "240"
pathLog = "<path_of_your_file>.txt"
debug = true
importMode = true

[config_soap]
user = "erp"
pass = "*****"
soapUrl="http://pfc.com/index.php/api/soap?wsdl=1"
```

Els paràmetres que s'han de configurar són els següents:

- **google_docs:**
email i *password* del compte on està el document Google Docs.
source: nom del projecte
fn: nom que tindrà el document/nom que té el document.
dictionary: direcció on està guardat el .csv que indica la relació columna – atribut
timeOut: temps que tenen les connexions SOAP per respondre abans que acabi la connexió.
pathLog: lloc on anirà el fitxer de log.
debug: si aquest valor està a true, es guardarà molta informació en el fitxer de debug, per exemple, quan s’inicia el programa, quins processos fa, etc.
importMode: si s’activa aquest valor, es farà un resum del que farà el programa abans d’executar totes les importacions. Ideal per l’execució en consola i no programada periòdicament.
- **config_soap:**
user i *password* per a la connexió SOAP.
soapUrl: Url on es troba el WSDL per la utilització del web service SOAP.

3. Un cop guardat el fitxer .ini. **L’administrador executarà el programa amb la següent instrucció: `php -f index.php GoogleImport.php <path del ini>.ini`.**

Amb aquesta instrucció (executada en consola Linux), es crearà el document en cas que no existeixi, amb totes les columnes que s’han especificat anteriorment al fitxer .csv. Aquesta instrucció es pot programar en un cron perquè s’executi cada cert temps però això encara no s’ha provat en entorns reals. Els paràmetres que s’especifiquen en la instrucció són:

Index.php: fitxer principal del programa que carrega diferents configuracions.

GoogleImport.php: nom del controlador. Recordem que hi ha l’opció d’executar altres controladors i, variant aquest paràmetre, el programa executarà un controlador o un altre.

.ini: fitxer amb els paràmetres de configuració que s’han especificat anteriorment.

4. Un cop s’ha creat el document, **L’administrador ja pot compartir el document amb el client que ha d’introduir les dades** (figura 24). En cas que el programa s’hagi configurat en un cron, el programa anirà important els productes a cada execució i anirà indicant al client si algun valor és incorrecte. En cas contrari, l’administrador esperarà a que el client hagi acabat o indiqui que ja es pot importar el que té.

Yobuyoutdoor-Productos ☆

Archivo Editar Ver Insertar Formato Datos Herramientas Ayuda

La última modificación la ha realizado info hace 3 días

\$

%

123

11pt

B

Abc

A

IA

Figura 24: Document Google Docs de productes

- Un cop el client ha indicat que ja es pot importar tot el contingut del document, **l'administrador tornarà a executar el programa amb la mateixa instrucció (php –f index.php GoogleImport.php <path del ini>.ini).**

Amb aquests passos s'importen o s'actualitzen els productes de Google a Magento.

Exemples de missatges d'error a Google Docs

Quan s'executa el programa, es fa una validació del contingut que ha introduït el client. Les validacions es fan pels conjunts d'atributs, tipus de productes, IVA, status i atributs configurables. Quan el programa detecta que hi ha un error, indica que aquest text que s'ha introduït és incorrecte i, a més, insereix una cadena de text que, si la torna a detectar en la següent execució, farà que no processi el producte fins que hagi arreglat aquest error. Es pot veure un exemple d'error a la figura 25.

descripcion-corta	cantidad-en-stock	peso	talla	associated	atributo-configurab
my desc.	400	2			
my desc.	400	2	#!ERROR: no s'ha trobat opcions per el atribut talla: XL		talla
my desc.	400	2	M		talla
descripcion-corta	100	5	L		talla

Figura 25: Error en un document de productes Google Docs. S'ha de destacar que el color verd de la cel·la és opcional i, en aquest cas, el document tenia una funció que buscava la paraula #!ERROR i, en cas de tenir aquest text, es canviava el color de la cel·la

La cadena de caràcters #!ERROR és la que parseja el programa per saber si el producte ja havia estat validat i si tenia algun error anterior.

Bibliografia

[api-google] Google Code, *API de Google Documents*, [web: Última visualització el 20/12/2011] <http://code.google.com/intl/ca/apis/documents/>

[api-google2] Google Code, *API de Google Spreadsheets*, [web: Última visualització el 21/12/2011] <http://code.google.com/intl/ca/apis/spreadsheets/>

[article] Les botigues online, *perspectiva d'un desenvolupament estratègic*. [web: Última visualització el 20/12/2012] <http://www.monografias.com/trabajos901/tiendas-online-desarrollo-estrategico-comunicacional/tiendas-online-desarrollo-estrategico-comunicacional.shtml>

[article2] El comerç Electronic, *Anàlisis del comerç electrònic* [web: Última visualització el 30/11/11] <http://www.monografias.com/trabajos15/comercio-electronico/comercio-electronico.shtml>

[drupal] Wikipedia, *Drupal* [web: Última visualització el 04/01/2012] <http://es.wikipedia.org/wiki/Drupal>

[drupal2] Drupal, *web oficial del CMS Drupal* [web: Última visualització el 04/01/2012] <http://drupal.org/>

[etailers] The Etailers, *pàgina oficial de l'empresa Etailers* [web: Última visualització el 04/01/2012] <http://www.theetailers.com>

[front-controller] Design patterns, *Patró front controller* [web: Última visualització el 21/09/2011] <http://java.sun.com/blueprints/patterns/FrontController.html>

[git] Documentació Git, *sistema de control de versions*, [web: Última visualització el 03/11/2011] <http://git-scm.com/documentation>

[joomla] Wikipedia, *Joomla* [web: Última visualització el 04/01/2012] <http://es.wikipedia.org/wiki/Joomla!>

[joomla2] Joomla, *web oficial del CMS Joomla* [web: Última visualització el 04/01/2012] <http://www.joomla.org/>

[magento] The Definitive Guide to Magento. *Adam McCombs and Robert Banh* [manual pdf].

[magento2] Magento Commerce, *eCommerce Plataforma for Growth. Traducció feta per David Parlori*. [manuak pdf i montat en PDF per Ivan Bejarano]

[magento3] Magento, *Pàgina oficial de la plataforma Magento*, [web: Última visualització el 19/12/2011] <http://www.magentocommerce.com/>

[magento4] Wikipedia. *Magento*. [web: Última visualització el 19/12/2011] <http://en.wikipedia.org/wiki/Magento>

[MVC] Wikipieda, *Patró Model Vista Controlador* [web: Última visualització el 14/09/2011] http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

[llei] Noticias Jurídiques, *Llei Orgànica de protecció de dades de caràcter personal* [web: Última visualització el 02/4/2011] http://noticias.juridicas.com/base_datos/Admin/lo15-1999.html

[llei2] LSSI, *Llei de serveis de la societat de la informació* [web: Última visualització el 15/12/11] <http://www.minetur.gob.es/dgdsi/lssi/Paginas/Index.aspx/>

[oscommerce] OsCommerce, *Pàgina oficial de la plataforma OsCommerce*, [web: Última visualització el 21/12/2011] <http://www.oscommerce.com/>

[php] PHP, *Pàgina oficial del llenguatge de programació PHP*, [web: Última visualització el 23/12/2011] <http://www.php.net/>

[prestashop] Prestashop, *Pàgina oficial de la plataforma Prestashop*, [web: Última visualització el 20/12/2011] <http://www.prestashop.com/en/>

[python] Python para todos. *Raúl Gonzalez Duque*. [manual pdf].

[singleton] Wikipieda, *Patró Singleton* [web: Última visualització el 14/09/2011] <http://es.wikipedia.org/wiki/Singleton>

[soap] Wikipedia, *Explicació del protocol SOAP*, [web: Última visualització el 22/11/2011] http://es.wikipedia.org/wiki/Simple_Object_Access_Protocol

[vim] VIM, *Pàgina oficial del editor VIM*, [web: Última visualització el 03/06/2011] <http://www.vim.org/>

[virtualenv] Python, *virtualenv*. [web: Última visualització el 30/08/2011] <http://pypi.python.org/pypi/virtualenv>

[xml-rpc] Wikipedia, *Explicació del protocol XML-RPC*, [web: Última visualització el 18/12/2011] <http://es.wikipedia.org/wiki/XML-RPC>

[zend-framework] Zend Framework, *Pàgina oficial de Zend Framework*, [web: Última visualització el 22/12/2011] <http://framework.zend.com/>

[zend-framework2] Manual Zend Framework, *Guia de referència per el programador*. [web:
Última visualització el 20/12/2011] <http://manual.zfdes.com/>

Resum:

Aquest projecte tracta de la millora de l'actual sistema d'importació que disposa Magento. Actualment per importar els productes a una botiga Magento es necessita treballar amb un fitxer csv on s'introdueix una gran quantitat de dades i requereix una gran quantitat de temps i esforç per part de l'empresa. Per millorar l'actual sistema, es proposa crear una integració entre Magento i Google Docs, fent que tots els productes que es vulguin importar s'introdueixin en un full de càlcul Google Docs i s'importin a la botiga Magento.

Resumen:

Este proyecto trata de la mejora del actual sistema de importación que contiene Magento. En la actualidad, para importar los productos a una tienda Magento se requiere trabajar con un fichero csv donde se introducen grandes cantidades de datos i requiere una gran cantidad de tiempo y esfuerzo por parte de la empresa. Para mejorar dicho sistema, se propone la creación de una integración entre Magento y Google Docs, haciendo que todos los productos que se desee importar se introduzcan en una hoja de cálculo Google Docs y se importen a la tienda Magento.

Abstract:

This project is about improving Magento's present importation system. Nowadays, to import products to a Magento store, it is necessary to work with a CSV file which contains a huge amount of data and it takes a lot of time and effort for the company. To improve the present system, it is proposed an integration between Magento and Google Docs. Thus, all products that are wished to import, will be introduced on a Google Docs calculus sheet and imported to Magento store.